Combinatorial optimisation for telecommunications
Lecture notes
Second part of the course

FRÉDÉRIC GIROIRE
frederic.giroire@cnrs.fr

Contents

Chapter 1

Linear programming

The nature of the programmes a computer scientist has to conceive often requires some knowledge in a specific domain of application, for example corporate management, network protocols, sound and video for multimedia streaming,... Linear programming is one of the necessary knowledges to handle optimization problems. These problems come from varied domains as production management, economics, transportation network planning, ... For example, one can mention the composition of train wagons, the electricity production, or the flight planning by airplane companies.

Most of these optimization problems do not admit an optimal solution that can be computed in a reasonable time, that is in polynomial time. However, we know how to efficiently solve some particular problems and to provide an optimal solution (or at least quantify the difference between the provided solution and the optimal value) by using techniques from linear programming.

In fact, in 1947, G.B. Dantzig conceived the Simplex Method to solve military planning problems asked by the US Air Force that were written as a linear programme, that is a system of linear equations. In this course, we introduce the basic concepts of linear programming. We then present the Simplex Method, following the book of V. Chvátal [2]. If you want to read more about linear programming, some good references are [6, 1].

The objective is to show the reader how to model a problem with a linear programme when it is possible, to present him different methods used to solve it or at least provide a good approximation of the solution. To this end, we present the *theory of duality* which provide ways of finding good bounds on specific solutions.

We also discuss the practical side of linear programming: there exist very efficient tools to solve linear programmes, e.g. CPLEX [3] and GLPK [4]. We present the different steps leading to the solution of a practical problem expressed as a linear programme.

1.1 Introduction

A *linear programme* is a problem consisting in maximizing or minimizing a linear function while satisfying a finite set of linear constraints.

Linear programmes can be written under the *standard form*:

$$\begin{array}{lrcll} \text{Maximize} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for all } 1 \leq i \leq m \\ & x_j & \geq & 0 & \text{for all } 1 \leq j \leq n. \end{array} \tag{1.1}$$

All constraints are inequalities (and not equations) and all variables are non-negative. The variables x_j are referred to as *decision variables*. The function that has to be maximized is called the problem *objective function*.

Observe that a constraint of the form $\sum_{j=1}^{n} a_{ij} x_j \geq b_i$ may be rewritten as $\sum_{j=1}^{n}(-a_{ij})x_j \leq -b_i$. Similarly, a minimization problem may be transformed into a maximization problem: minimizing $\sum_{j=1}^{n} c_j x_j$ is equivalent to maximizing $\sum_{j=1}^{n}(-c_j)x_j$. Hence, every maximization or minimization problem subject to linear constraints can be reformulated in the standard form (See Exercices **??** and **??**.).

A n-tuple $(x_1,\dots,x_n)$ satisfying the constraints of a linear programme is a *feasible solution* of this problem. A solution that maximizes the objective function of the problem is called an *optimal solution*. Beware that a linear programme does not necessarily admits a unique optimal solution. Some problems have several optimal solutions while others have none. The later case may occur for two opposite reasons: either there exist no feasible solutions, or, in a sense, there are too many. The first case is illustrated by the following problem.

$$\begin{array}{lrcrcr} \text{Maximize} & 3x_1 & - & x_2 & & \\ \text{Subject to:} & x_1 & + & x_2 & \leq & 2 \\ & -2x_1 & - & 2x_2 & \leq & -10 \\ & & & x_1, x_2 & \geq & 0 \end{array} \tag{1.2}$$

which has no feasible solution (See Exercise 4). Problems of this kind are referred to as *unfeasible*. At the opposite, the problem

$$\begin{array}{lrcrcr} \text{Maximize} & x_1 & - & x_2 & & \\ \text{Subject to:} & -2x_1 & + & x_2 & \leq & -1 \\ & -x_1 & - & 2x_2 & \leq & -2 \\ & & & x_1, x_2 & \geq & 0 \end{array} \tag{1.3}$$

has feasible solutions. But none of them is optimal (See Exercise 4). As a matter of fact, for every number M, there exists a feasible solution x_1, x_2 such that $x_1 - x_2 > M$. The problems verifying this property are referred to as *unbounded*. Every linear programme satisfies exactly one the following assertions: either it admits an optimal solution, or it is unfeasible, or it is unbounded.

Geometric interpretation.

The set of points in $\mathbb{R}^n$ at which any single constraint holds with equality is a hyperplane in $\mathbb{R}^n$. Thus each constraint is satisfied by the points of a closed half-space of $\mathbb{R}^n$, and the set of feasible solutions is the intersection of all these half-spaces, a convex polyhedron P.

Because the objective function is linear, its level sets are hyperplanes. Thus, if the maximum value of $\mathbf{cx}$ over P is z^*, the hyperplane $\mathbf{cx} = z^*$ is a supporting hyperplane of P. Hence $\mathbf{cx} = z^*$ contains an extreme point (a corner) of P. It follows that the objective function attains its maximum at one of the extreme points of P.

1.2 The Simplex Method

The authors advise you, in a humanist élan, to skip this section if you are not ready to suffer. In this section, we present the principle of the Simplex Method. We consider here only the most general case and voluntarily omit here the degenerate cases to focus only on the basic principle. A more complete presentation can be found for example in [2].

1.2.1 A first example

We illustrate the Simplex Method on the following example:

$$\begin{array}{lrcrcrcr}
\text{Maximize} & 5x_1 & + & 4x_2 & + & 3x_3 & & \\
\text{Subject to:} & & & & & & & \\
 & 2x_1 & + & 3x_2 & + & x_3 & \leq & 5 \\
 & 4x_1 & + & x_2 & + & 2x_3 & \leq & 11 \\
 & 3x_1 & + & 4x_2 & + & 2x_3 & \leq & 8 \\
 & & & & & x_1, x_2, x_3 & \geq & 0.
\end{array} \tag{1.4}$$

The first step of the Simplex Method is to introduce new variables called *slack variables*. To justify this approach, let us look at the first constraint,

$$2x_1 + 3x_2 + x_3 \leq 5. \tag{1.5}$$

For all feasible solution x_1, x_2, x_3, the value of the left member of (1.5) is at most the value of the right member. But, there often is a gap between these two values. We note this gap x_4. In other words, we define $x_4 = 5 - 2x_1 - 3x_2 - x_3$. With this notation, Equation (1.5) can now be written as $x_4 \geq 0$. Similarly, we introduce the variables x_5 and x_6 for the two other constraints of Problem (1.4). Finally, we use the classic notation z for the objective function $5x_1 + 4x_2 + 3x_3$. To summarize, for all choices of x_1, x_2, x_3 we define x_4, x_5, x_6 and z by the formulas

$$\begin{array}{rcrcrcrcr}
x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\
x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\
x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\
z & = & & & 5x_1 & + & 4x_2 & + & 3x_3.
\end{array} \tag{1.6}$$

With these notations, the problem can be written as:

$$\text{Maximize } z \text{ subject to } x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. \tag{1.7}$$

The new variables that were introduced are referred as *slack variables*, when the initial variables are usually called the *decision variables*. It is important to note that Equation (1.6) define an equivalence between (1.4) and (1.7). More precisely:

- Any feasible solution (x_1, x_2, x_3) of (1.4) can be uniquely extended by (1.6) into a feasible solution $(x_1, x_2, x_3, x_4, x_5, x_6)$ of (1.7).

- Any feasible solution $(x_1,x_2,x_3,x_4,x_5,x_6)$ of (1.7) can be reduced by a simple removal of the slack variables into a feasible solution (x_1,x_2,x_3) of (1.4).
- This relationship between the feasible solutions of (1.4) and the feasible solutions of (1.7) allows to produce the optimal solution of (1.4) from the optimal solutions of (1.7) and *vice versa*.

The Simplex strategy consists in finding the optimal solution (if it exists) by successive improvements. If we have found a feasible solution (x_1,x_2,x_3) of (1.7), then we try to find a new solution $(\bar{x_1},\bar{x_2},\bar{x_3})$ which is better in the sense of the objective function:

$$5\bar{x_1}+4\bar{x_2}+3\bar{x_3} \geq 5x_1+4x_2+3x_3.$$

By repeating this process, we obtain at the end an optimal solution.

To start, we first need a feasible solution. To find one in our example, it is enough to set the decision variables x_1,x_2,x_3 to zero and to evaluate the slack variables x_4,x_5,x_6 using (1.6). Hence, our initial solution,

$$x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 5, x_5 = 11, x_6 = 8 \tag{1.8}$$

gives the result $z = 0$.

We now have to look for a new feasible solution which gives a larger value for z. Finding such a solution is not hard. For example, if we keep $x_2 = x_3 = 0$ and increase the value of x_1, then we obtain $z = 5x_1 \geq 0$. Hence, if we keep $x_2 = x_3 = 0$ and if we set $x_1 = 1$, then we obtain $z = 5$ (and $x_4 = 3, x_5 = 7, x_6 = 5$). A better solution is to keep $x_2 = x_3 = 0$ and to set $x_1 = 2$; we then obtain $z = 10$ (and $x_4 = 1, x_5 = 3, x_6 = 2$). However, if we keep $x_2 = x_3 = 0$ and if we set $x_1 = 3$, then $z = 15$ and $x_4 = x_5 = x_6 = -1$, breaking the constraint $x_i \geq 0$ for all i. The conclusion is that one can not increase x_1 as much as one wants. The question then is: how much can x_1 be raised (when keeping $x_2 = x_3 = 0$) while satisfying the constraints ($x_4,x_5,x_6 \geq 0$)?

The condition $x_4 = 5-2x_1-3x_2-x_3 \geq 0$ implies $x_1 \leq \frac{5}{2}$. Similarly, $x_5 \geq 0$ implies $x_1 \leq \frac{11}{4}$ and $x_6 \geq 0$ implies $x_1 \leq \frac{8}{3}$. The first bound is the strongest one. Increasing x_1 to this bound gives the solution of the next step:

$$x_1 = \frac{5}{2}, x_2 = 0, x_3 = 0, x_4 = 0, x_5 = 1, x_6 = \frac{1}{2} \tag{1.9}$$

which gives a result $z = \frac{25}{2}$ improving the last value $z = 0$ of (1.8).

Now, we have to find a new feasible solution that is better than (1.9). However, this task is not as simple as before. Why? As a matter of fact, we had at disposal the feasible solution of (1.8), but also the system of linear equations (1.6) which led us to a better feasible solution. Thus, we should build a new system of linear equations related to (1.9) in the same way as (1.6) is related to (1.8).

Which properties should have this new system? Note first that (1.6) express the strictly positive variables of (1.8) in function of the null variables. Similarly, the new system has to express the strictly positive variables of (1.9) in function of the null variables of (1.9): x_1,x_5,x_6 (and z) in function of x_2,x_3 and x_4. In particular, the variable x_1, whose value just increased from zero to a strictly positive value, has to go to the left side of the new system. The variable x_4, which is now null, has to take the opposite move.

To build this new system, we start by putting x_1 on the left side. Using the first equation of (1.6), we write x_1 in function of x_2, x_3, x_4:

$$x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \tag{1.10}$$

Then, we express x_5, x_6 and z in function of x_2, x_3, x_4 by substituting the expression of x_1 given by (1.10) in the corresponding lines of (1.6).

$$\begin{aligned}
x_5 &= 11 - 4\left(\frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4\right) - x_2 - 2x_3 \\
&= 1 + 5x_2 + 2x_4, \\
x_6 &= 8 - 3\left(\frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4\right) - 4x_2 - 2x_3 \\
&= \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4, \\
z &= 5\left(\frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4\right) + 4x_2 + 3x_3 \\
&= \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4.
\end{aligned}$$

So the new system is

$$\begin{array}{rcrcrcrcrcrcr}
x_1 & = & \frac{5}{2} & - & \frac{3}{2} & x_2 & - & \frac{1}{2} & x_3 & - & \frac{1}{2} & x_4 \\
x_5 & = & 1 & + & 5 & x_2 & & & & + & 2 & x_4 \\
x_6 & = & \frac{1}{2} & + & \frac{1}{2} & x_2 & - & \frac{1}{2} & x_3 & + & \frac{3}{2} & x_4 \\
z & = & \frac{25}{2} & - & \frac{7}{2} & x_2 & + & \frac{1}{2} & x_3 & - & \frac{5}{2} & x_4.
\end{array} \tag{1.11}$$

As done at the first iteration, we now try to increase the value of z by increasing a right variable of the new system, while keeping the other right variables at zero. Note that raising x_2 or x_4 would lower the value of z, against our objective. So we try to increase x_3. How much? The answer is given by (1.11) : with $x_2 = x_4 = 0$, the constraint $x_1 \geq 0$ implies $x_3 \leq 5$, $x_5 \geq 0$ impose no restriction and $x_6 \geq 0$ implies that $x_3 \leq 1$. To conclude $x_3 = 1$ is the best we can do, and the new solution is

$$x_1 = 2, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 0 \tag{1.12}$$

and the value of z increases from 12.5 to 13. As stated, we try to obtain a better solution but also a system of linear equations associated to (1.12). In this new system, the (strictly) positive variables x_2, x_4, x_6 have to appear on the right. To build this new system, we start by handling the new left variable, x_3. Thanks to the third equation of (1.11) we rewrite x_3 and by substitution in the remaining equations of (1.11) we obtain:

$$\begin{array}{rcrcrcrcr} x_3 & = & 1 & + & x_2 & + & 3x_4 & - & 2x_6 \\ x_1 & = & 2 & - & 2x_2 & - & 2x_4 & + & x_6 \\ x_5 & = & 1 & + & 5x_2 & + & 2x_4 & & \\ z & = & 13 & - & 3x_2 & - & x_4 & - & x_6. \end{array} \tag{1.13}$$

It is now time to do the third iteration. First, we have to find a variable of the right side of (1.13) whose increase would result in an increase of the objective z. But there is no such variable, as any increase of x_2, x_4 or x_6 would lower z. We are stuck. In fact, this deadlock indicates that the last solution is optimal. Why? The answer lies in the last line of (1.13):

$$z = 13 - 3x_2 - x_4 - x_6. \tag{1.14}$$

The last solution (1.12) gives a value $z = 13$; proving that this solution is optimal boils down to prove that any feasible solution satisfies $z \leq 13$. As any feasible solution $x_1, x_2, \ldots, x_6$ satisfies the inequalities $x_2 \geq 0, x_4 \geq 0, x_6 \geq 0$, then $z \leq 13$ directly derives from (1.14).

1.2.2 The dictionaries

More generally, given a problem

$$\begin{array}{llcll} \text{Maximize} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for all } 1 \leq i \leq m \\ & x_j & \geq & 0 & \text{for all } 1 \leq j \leq n \end{array} \tag{1.15}$$

we first introduce the *slack variables* $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ and we note the objective function z. That is, we define

$$\begin{array}{rcll} x_{n+i} & = & b_i - \sum_{j=1}^{n} a_{ij} x_j & \text{for all } 1 \leq i \leq m \\ z & = & \sum_{j=1}^{n} c_j x_j & \end{array} \tag{1.16}$$

In the framework of the Simplex Method, each feasible solution $(x_1, x_2, \ldots, x_n)$ of (1.15) is represented by $n+m$ positive or null numbers $x_1, x_2, \ldots, x_{n+m}$, with $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ defined by (1.16). At each iteration, the Simplex Method goes from one feasible solution $(x_1, x_2, \ldots, x_{n+m})$ to an other feasible solution $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{n+m})$, which is better in the sense that

$$\sum_{j=1}^{n} c_j \bar{x}_j > \sum_{j=1}^{n} c_j x_j.$$

As we have seen in the example, it is convenient to associate a system of linear equations to each feasible solution. As a matter of fact, it allows to find better solutions in an easy way. The technique is to translate the choices of the values of the variables of the right side of the system into the variables of the left side and in the objective function as well. These systems have been named *dictionaries* by J.E. Strum (1972). Thus, every dictionary associated to (1.15) is a system of equations whose variables $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ and z are expressed in function of $x_1, x_2, \ldots, x_n$. These $n+m+1$ variables are closely linked and every dictionary express these dependencies.

Property 1.1. *Any feasible solution of the equations of a dictionary is also a feasible solution of (1.16) and* vice versa*.*

For example, for any choice of $x_1, x_2, \ldots, x_6$ and of z, the three following assertions are equivalent:

- $(x_1, x_2, \ldots, x_6, z)$ is a feasible solution of (1.6);
- $(x_1, x_2, \ldots, x_6, z)$ is a feasible solution of (1.11);
- $(x_1, x_2, \ldots, x_6, z)$ is a feasible solution of (1.13).

From this point of view, the three dictionaries (1.6), (1.11) and (1.13) contain the same information on the dependencies between the seven variables. However, each dictionary present this information in a specific way. (1.6) suggests that the values of the variables x_1, x_2 and x_3 can be chosen at will while the values of x_4, x_5, x_6 and z are fixed. In this dictionary, the decision variables x_1, x_2, x_3 act as independent variables while the slack variables x_4, x_5, x_6 are related to each other. In the dictionary (1.13), the independent variables are x_2, x_4, x_6 and the related ones are x_3, x_1, x_5, z.

Property 1.2. *The equations of a dictionary have to express m variables among $x_1, x_2, \ldots, x_{n+m}, z$ in function of the n remaining others.*

Properties 1.1 and 1.2 define what a dictionary is. In addition to these two properties, the dictionaries (1.6),(1.11) and (1.13) have the following property.

Property 1.3. *When putting the right variables to zero, one obtains a feasible solution by evaluating the left variables.*

The dictionaries that have this last property are called *feasible dictionaries*. As a matter of fact, any feasible dictionary describes a feasible solution. However, all feasible solutions cannot be described by a feasible dictionary. For example, no dictionary describe the feasible solution $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 2$, $x_5 = 5$, $x_6 = 3$ of (1.4). The feasible solutions that can be described by dictionaries are referred as *basic solutions*. The Simplex Method explores only basic solutions and ignores all other ones. But this is valid because if an optimal solution exists, then there is an optimal and basic solution. Indeed, if a feasible solution cannot be improved by the Simplex Method, then increasing any of the n right variables to a positive value never increases the objective function. In such case, the objective function must be written as a linear function of these variables in which all the coefficient are non-positive, and thus the objective function is clearly maximum when all the right variables equal zero. For example, it was the case in (1.14).

1.2.3 Finding an initial solution

In the previous examples, the initialization of the Simplex Method was not a problem. As a matter of fact, we carefully chose problems with all b_i non-negative. This way $x_1 = 0$, $x_2 = 0$,

$\cdots, x_n = 0$ was a feasible solution and the dictionary was easily built. These problems are called *problems with a feasible origin*.

What happens when confronted with a problem with an unfeasible origin? Two difficulties arise. First, a feasible solution can be hard to find. Second, even if we find a feasible solution, a feasible dictionary has then to be built. A way to solve these difficulties is to use another problem called *auxiliary problem*:

$$\begin{array}{lrcll} \text{Minimise} & x_0 & & & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij}x_j - x_0 & \leq & b_i & (i = 1,2,\cdots,m) \\ & x_j & \geq & 0 & (j = 0,1,\cdots,n). \end{array}$$

A feasible solution of the auxiliary problem is easily available: it is enough to set $x_j = 0 \forall j \in [1 \ldots n]$ and to give to x_0 a big enough value. It is now easy to see that the original problem has a feasible solution if and only if the auxiliary problem has a feasible solution with $x_0 = 0$. In other words, the original problem has a feasible solution if the optimal value of the auxiliary problem is null. Thus, the idea is to first solve the auxiliary problem. Let see the details on an example.

$$\begin{array}{lrcrcrcr} \text{Maximise} & x_1 & - & x_2 & + & x_3 & & \\ \text{Subject to :} & & & & & & & \\ & 2x_1 & - & x_2 & + & 2x_3 & \leq & 4 \\ & 2x_1 & - & 3x_2 & + & x_3 & \leq & -5 \\ & -x_1 & + & x_2 & - & 2x_3 & \leq & -1 \\ & & & & & x_1,x_2,x_3 & \geq & 0 \end{array}$$

$$\begin{array}{lrcrcrcrcr} \text{Maximise} & -x_0 & & & & & & & & \\ \text{Subject to:} & & & & & & & & & \\ & 2x_1 & - & x_2 & + & 2x_3 & - & x_0 & \leq & 4 \\ & 2x_1 & - & 3x_2 & + & x_3 & - & x_0 & \leq & -5 \\ & -x_1 & + & x_2 & - & 2x_3 & - & x_0 & \leq & -1 \\ & & & & & & & x_1,x_2,x_3,x_0 & \geq & 0 \end{array}$$

We introduce the slack variables. We obtain the dictionary:

$$\begin{array}{rcrcrcrcrcr} x_4 & = & 4 & - & 2x_1 & + & x_2 & - & 2x_3 & + & x_0 \\ x_5 & = & -5 & - & 2x_1 & + & 3x_2 & - & x_3 & + & x_0 \\ x_6 & = & -1 & + & x_1 & - & x_2 & + & 2x_3 & + & x_0 \\ w & = & & & & & & & & - & x_0. \end{array} \tag{1.17}$$

Note that this dictionary is not feasible. However it can be transformed into a feasible one by operating a simple pivot , x_0 entering the basis as x_5 exits it:

$$\begin{array}{rcrcrcrcrcr} x_0 & = & 5 & + & 2x_1 & - & 3x_2 & + & x_3 & + & x_5 \\ x_4 & = & 9 & & & - & 2x_2 & - & x_3 & + & x_5 \\ x_6 & = & 4 & +3 & x_1 & - & 4x_2 & + & 3x_3 & + & x_5 \\ \hline w & = & -5 & - & 2x_1 & + & 3x_2 & - & x_3 & - & x_5. \end{array}$$

More generally, the auxiliary problem can be written as

$$\begin{array}{lrcll} \text{Maximise} & -x_0 & & & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij}x_j - x_0 & \leq & b_i & (i=1,2,\cdots,m) \\ & x_j & \geq & 0 & (j=0,1,2,\cdots,n) \end{array}$$

and the associated dictionary is

$$\begin{array}{rclcll} x_{n+i} & = & b_i - & \sum_{j=1}^{n} a_{ij}x_j & + \; x_0 & (i=1,2,\cdots,m) \\ w & = & & & - \; x_0 & \end{array}$$

This dictionary can be made feasible by pivoting x_0 with the variable the "most unfeasible", that is the exiting variable x_{n+k} is the one with $b_k \leq b_i$ for all i. After the pivot, the variable x_0 has value $-b_k$ and each x_{n+i} has value $b_i - b_k$. All these values are non negative. We are now able to solve the auxiliary problem using the simplex method. Let us go back to our example.

After the first iteration with x_2 entering and x_6 exiting, we get:

$$\begin{array}{rcrcrcrcrcr} x_2 & = & 1 & + & 0.75x_1 & + & 0.75x_3 & + & 0.25x_5 & - & 0.25x_6 \\ x_0 & = & 2 & - & 0.25x_1 & - & 1.25x_3 & + & 0.25x_5 & + & 0.75x_6 \\ x_4 & = & 7 & - & 1.5x_1 & - & 2.5x_3 & + & 0.5x_5 & + & 0.5x_6 \\ \hline w & = & -2 & + & 0.25x_1 & + & 1.25x_3 & - & 0.25x_5 & - & 0.75x_6. \end{array}$$

After the second iteration with x_3 entering and x_0 exiting:

$$\begin{array}{rcrcrcrcrcr} x_3 & = & 1.6 & - & 0.2x_1 & + & 0.2x_5 & + & 0.6x_6 & - & 0.8x_0 \\ x_2 & = & 2.2 & + & 0.6x_1 & + & 0.4x_5 & + & 0.2x_6 & - & 0.6x_0 \\ x_4 & = & 3 & - & x_1 & & & - & x_6 & + & 2x_0 \\ \hline w & = & & & & & & & & - & x_0. \end{array} \tag{1.18}$$

The last dictionary (1.18) is optimal. As the optimal value of the auxiliary problem is null, this dictionary provides a feasible solution of the original problem: $x_1 = 0, x_2 = 2.2, x_3 = 1.6$. Moreover, (1.18) can be easily transformed into a feasible dictionary of the original problem. To obtain the first three lines of the desired dictionary, it is enough to copy the first three lines while removing the terms with x_0:

$$\begin{array}{rcrcrcrcr} x_3 & = & 1.6 & - & 0.2x_1 & + & 0.2x_5 & + & 0.6x_6 \\ x_2 & = & 2.2 & + & 0.6x_1 & + & 0.4x_5 & + & 0.2x_6 \\ x_4 & = & 3 & - & x_1 & & & - & x_6 \end{array} \tag{1.19}$$

To obtain the last line, we express the original objective function

$$z = x_1 - x_2 + x_3 \tag{1.20}$$

in function of the variables outside the basis x_1, x_5, x_6. To do so, we replace the variables of (1.20) by (1.19) and we get:

$$z = x_1 - (2.2 + 0.6x_1 + 0.4x_5 + 0.2x_6) + (1.6 - 0.2x_1 + 0.2x_5 + 0.6x_6) \tag{1.21}$$
$$z = -0.6 + 0.2x_1 - 0.2x_5 + 0.4x_6$$

The desired dictionary then is:

$$\begin{array}{rcrcrcrcr}
x_3 & = & 1.6 & - & 0.2x_1 & + & 0.2x_5 & + & 0.6x_6 \\
x_2 & = & 2.2 & + & 0.6x_1 & + & 0.4x_5 & + & 0.2x_6 \\
x_4 & = & 3 & - & x_1 & & & - & x_6 \\
\hline
z & = & -0.6 & + & 0.2x_1 & - & 0.2x_5 & + & 0.4x_6
\end{array}$$

This strategy is known as the *Simplex Method in two phases*. During the first phase, we set and solve the auxiliary problem. If the optimal value is null, we do the second phase consisting in solving the original problem. Otherwise, the original problem is not feasible.

1.3 Duality of linear programming

Any maximization linear programme has a corresponding minimization problem called the *dual problem*. Any feasible solution of the dual problem gives an upper bound on the optimal value of the initial problem, which is called the *primal*. Reciprocally, any feasible solution of the primal provides a lower bound on the optimal value of the dual problem. Actually, if one of both problems admits an optimal solution, then the other problem does as well and the optimal solutions match each other. This section is devoted to this result also known as the *Duality Theorem*. Another interesting application of the dual problem is that, in some problems, the variables of the dual have some useful interpretation.

1.3.1 Motivations: providing upper bounds on the optimal value

A way to quickly estimate the optimal value of a maximization linear programme simply consists in computing a feasible solution whose value is sufficiently large. For instance, let us consider the following problem formulated in Problem 1.4. The solution $(0,0,1,0)$ gives us a lower bound of 5 for the optimal value z^*. Even better, we get $z^* \geq 22$ by considering the solution $(3,0,2,0)$. Of course, doing so, we have no way to know how close to the optimal value the computed lower bound is.

Problem 1.4.

$$\begin{array}{rrcl}
\text{Maximize} & 4x_1+x_2+5x_3+3x_4 & & \\
\text{Subject to:} & x_1-x_2-x_3+3x_4 & \leq & 1 \\
 & 5x_1+x_2+3x_3+8x_4 & \leq & 55 \\
 & -x_1+2x_2+3x_3-5x_4 & \leq & 3 \\
 & x_1,x_2,x_3,x_4 & \geq & 0
\end{array}$$

The previous approach provides lower bounds on the optimal value. However, this intuitive method is obviously less efficient than the Simplex Method and this approach provides no clue about the optimality (or not) of the obtained solution. To do so, it is interesting to have upper bounds on the optimal value. This is the main topic of this section.

How to get an upper bound for the optimal value in the previous example? A possible approach is to consider the constraints. For instance, multiplying the second constraint by $\frac{5}{3}$, we get that $z^* \leq \frac{275}{3}$. Indeed, for any $x_1, x_2, x_3, x_4 \geq 0$:

$$\begin{aligned} 4x_1 + x_2 + 5x_3 + 3x_4 &\leq \frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4 = (5x_1 + x_2 + 3x_3 + 8x_4) \times \frac{5}{3} \\ &\leq 55 \times \frac{5}{3} = \frac{275}{3} \end{aligned}$$

In particular, the above inequality is satisfied by any optimal solution. Therefore, $z^* \leq \frac{275}{3}$. Let us try to improve this bound. For instance, we can add the second constraint to the third one. This gives, for any $x_1, x_2, x_3, x_4 \geq 0$:

$$\begin{aligned} 4x_1 + x_2 + 5x_3 + 3x_4 &\leq 4x_1 + 3x_2 + 6x_3 - 3x_4 \\ &\leq (5x_1 + x_2 + 3x_3 + 8x_4) + (-x_1 + 2x_2 + 3x_3 - 5x_4) \\ &\leq 55 + 3 = 58 \end{aligned}$$

Hence, $z^* \leq 58$.

More formally, we try to upper bound the optimal value by a linear combination of the constraints. Precisely, for all i, let us multiply the i^{th} constraint by $y_i \geq 0$ and then sum the resulting constraints. In the previous two examples, we had $(y_1, y_2, y_3) = (0, \frac{5}{3}, 0)$ and $(y_1, y_2, y_3) = (0, 1, 1)$. More generally, we obtain the following inequality:

$$\begin{aligned} & y_1(x_1 - x_2 - x_3 + 3x_4) + y_2(5x_1 + x_2 + 3x_3 + 8x_4) + y_3(-x_1 + 2x_2 + 3x_3 - 5x_4) \\ = \; & (y_1 - 5y_2 - y_3)x_1 + (-y_1 + y_2 + 2y_3)x_2 + (-y_1 + 3y_2 + 3y_3)x_3 + (3y_1 + 8y_2 - 5y_3)x_4 \\ \leq \; & y_1 + 55y_2 + 3y_3 \end{aligned}$$

For this inequality to provide an upper bound of $4x_1 + x_2 + 5x_3 + 3x_4$, we need to ensure that, for all $x_1, x_2, x_3, x_4 \geq 0$,

$$\begin{aligned} & 4x_1 + x_2 + 5x_3 + 3x_4 \\ \leq \; & (y_1 - 5y_2 - y_3)x_1 + (-y_1 + y_2 + 2y_3)x_2 + (-y_1 + 3y_2 + 3y_3)x_3 + (3y_1 + 8y_2 - 5y_3)x_4. \end{aligned}$$

That is, $y_1 - 5y_2 - y_3 \geq 4$, $-y_1 + y_2 + 2y_3 \geq 1$, $-y_1 + 3y_2 + 3y_3 \geq 5$, and $3y_1 + 8y_2 - 5y_3 \geq 3$.

Combining all inequalities, we obtain the following minimization linear programme:

$$\begin{array}{rl} \text{Minimize} & y_1 + 55y_2 + 3y_3 \\ \text{Subject to:} & \\ & \begin{array}{rcl} y_1 - 5y_2 - y_3 & \geq & 4 \\ -y_1 + y_2 + 2y_3 & \geq & 1 \\ -y_1 + 3y_2 + 3y_3 & \geq & 5 \\ 3y_1 + 8y_2 - 5y_3 & \geq & 3 \\ y_1, y_2, y_3 & \geq & 0 \end{array} \end{array}$$

This problem is called the *dual* of the initial maximization problem.

1.3.2 Dual problem

We generalize the example given in Subsection 1.3.1. Consider the following general maximization linear programme:

Problem 1.5.

$$\begin{array}{rll} \text{Maximize} & \sum_{j=1}^{n} c_j x_j & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & \text{for all } 1 \leq i \leq m \\ & x_j \geq 0 & \text{for all } 1 \leq j \leq n \end{array}$$

Problem 1.15 is called the *primal*. The matricial formulation of this problem is

$$\begin{array}{rl} \text{Maximize} & \mathbf{c}^T \mathbf{x} \\ \text{Subject to:} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & x \geq \mathbf{0} \end{array}$$

where $\mathbf{x}^T = [x_1, \ldots, x_n]$ and $\mathbf{c}^T = [c_1, \ldots, c_n]$ are vectors in $\mathbb{R}^n$, and $\mathbf{b}^T = [b_1, \ldots, b_m] \in \mathbb{R}^m$, and $\mathbf{A} = [a_{ij}]$ is a matrix in $\mathbb{R}^{m \times n}$.

To find an upper bound on $\mathbf{c}^T\mathbf{x}$, we aim at finding a vector $\mathbf{y}^T = [y_1, \ldots, y_m] \geq 0$ such that, for all feasible solutions $\mathbf{x} \geq 0$ of the initial problem, $\mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T\mathbf{A}\mathbf{x} \leq \mathbf{y}^T\mathbf{b} = \mathbf{b}^T\mathbf{y}$, that is:

$$\begin{array}{rl} \text{Minimize} & \mathbf{b}^T \mathbf{y} \\ \text{Subject to:} & \mathbf{A}^T\mathbf{y} \geq \mathbf{c} \\ & \mathbf{y} \geq \mathbf{0} \end{array}$$

In other words, the *dual* of Problem 1.15 is defined by:

Problem 1.6.

$$\begin{array}{rll} \text{Minimize} & \sum_{i=1}^{m} b_i y_i & \\ \text{Subject to:} & \sum_{i=1}^{m} a_{ij} y_i \geq c_j & \text{for all } 1 \leq j \leq n \\ & y_i \geq 0 & \text{for all } 1 \leq i \leq m \end{array}$$

Notice that the dual of a maximization problem is a minimization problem. Moreover, there is a one-to-one correspondence between the m constraints of the primal $\sum_{j=1\ldots n} a_{ij} x_j \leq b_i$ and the m variables y_i of the dual. Similarly, the n constraints $\sum_{i=1}^{m} a_{ij} y_i \geq c_j$ of the dual correspond one-to-one to the n variables x_i of the primal.

Problem 1.16, which is the dual of Problem 1.15, can be equivalently formulated under the standard form as follows.

$$\begin{array}{rll} \text{Maximize} & \sum_{i=1}^{m} (-b_i) y_i & \\ \text{Subject to:} & \sum_{i=}^{m} (-a_{ij}) y_i \leq -c_j & \text{for all } 1 \leq j \leq n \\ & y_i \geq 0 & \text{for all } 1 \leq i \leq m \end{array} \tag{1.22}$$

Then, the dual of Problem 1.22 has the following formulation which is equivalent to Problem 1.15.

$$\begin{array}{rll} \text{Minimize} & \sum_{j=1}^{n} (-c_j) x_j & \\ \text{Subject to:} & \sum_{j=1}^{n} (-a_{ij}) x_j \geq -b_i & \text{for all } 1 \leq i \leq m \\ & x_j \geq 0 & \text{for all } 1 \leq j \leq n \end{array} \tag{1.23}$$

We deduce the following lemma.

Lemma 1.7. *If D is the dual of a problem P, then the dual of D is P. Informally, the dual of the dual is the primal.*

1.3.3 Duality Theorem

An important aspect of duality is that feasible solutions of the primal and the dual are related.

Lemma 1.8. *Any feasible solution of Problem 1.16 yields an upper bound for Problem 1.15. In other words, the value given by any feasible solution of the dual of a problem is an upper bound for the primal problem.*

Proof. Let $(y_1,\dots,y_m)$ be a feasible solution of the dual and $(x_1,\dots,x_n)$ be a feasible solution of the primal. Then,

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

□

Corollary 1.9. *If $(y_1,\dots,y_m)$ is a feasible solution of the dual of a problem (Problem 1.16) and $(x_1,\dots,x_n)$ is a feasible solution of the corresponding primal (Problem 1.15) such that $\sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i$, then both solutions are optimal.*

Corollary 1.9 states that if we find two solutions for the dual and the primal achieving the same value, then this is a certificate of the optimality of these solutions. In particular, in that case (if they are feasible), both the primal and the dual problems have same optimal value.

For instance, we can easily verify that $(0,14,0,5)$ is a feasible solution for Problem 1.4 with value 29. On the other hand, $(11,0,6)$ is a feasible solution for the dual with same value. Hence, the optimal solutions for the primal and for the dual coincide and are equal to 29.

In general, it is not immediate that any linear programme may have such certificate of optimality. In other words, for any feasible linear programme, can we find a solution of the primal problem and a solution of the dual problem that achieve the same value (thus, this value would be optimal)? One of the most important result of the linear programming is the duality theorem that states that it is actually always the case: for any feasible linear programme, the primal and the dual problems have the same optimal solution. This theorem has been proved by D. Gale, H.W. Kuhn and A. W. Tucker [5] and comes from discussions between G.B. Dantzig and J. von Neumann during Fall 1947.

Theorem 1.10 (DUALITY THEOREM). *If the primal problem defined by Problem 1.15 admits an optimal solution $(x_1^*,\dots,x_n^*)$, then the dual problem (Problem 1.16) admits an optimal solution $(y_1^*,\dots,y_m^*)$, and*

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*.$$

Proof. The proof consists in showing how a feasible solution $(y_1^*, \ldots, y_m^*)$ of the dual can be obtained thanks to the Simplex Method, so that $z^* = \sum_{i=1}^m b_i y_i^*$ is the optimal value of the primal. The result then follows from Lemma 1.8.

Let us assume that the primal problem has been solved by the Simplex Method. For this purpose, the slack variables have been defined by

$$x_{n+i} = b_i - \sum_{j=1}^n a_{ij} x_j \quad \text{for } 1 \le i \le m.$$

Moreover, the last line of the last dictionary computed during the Simplex Method gives the optimal value z^* of the primal in the following way: for any feasible solution $(x_1, \ldots, x_n)$ of the primal we have

$$z = \sum_{j=1}^n c_j x_j = z^* + \sum_{i=1}^{n+m} \bar{c}_i x_i.$$

Recall that, for all $i \le n+m$, $\bar{c}_i$ is non-positive, and that it is null if x_i is one of the basis variables. We set

$$y_i^* = -\bar{c}_{n+i} \qquad \text{for } 1 \le i \le m.$$

Then, by definition of the y_i^*'s and the x_{n+i}'s for $1 \le i \le m$, we have

$$\begin{aligned} z = \sum_{j=1}^n c_j x_j &= z^* + \sum_{i=1}^n \bar{c}_i x_i - \sum_{i=1}^m y_i^* \left(b_i - \sum_{j=1}^n a_{ij} x_j \right) \\ &= \left(z^* - \sum_{i=1}^m y_i^* b_i \right) + \sum_{j=1}^n \left(\bar{c}_j + \sum_{i=1}^m a_{ij} y_i^* \right) x_j. \end{aligned}$$

Since this equation must be true whatever be the affectation of the x_i's and since the $\bar{c}_i$'s are non-positive, this leads to

$$\begin{aligned} z^* &= \sum_{i=1}^m y_i^* b_i \quad \text{and} \\ c_j &= \bar{c}_j + \sum_{i=1}^m a_{ij} y_i^* \le \sum_{i=1}^m a_{ij} y_i^* \quad \text{for all } 1 \le j \le n. \end{aligned}$$

Hence, $(y_1^*, \ldots, y_m^*)$ defined as above is a feasible solution achieving the optimal value of the primal. By Lemma 1.8, this is an optimal solution of the dual. □

1.3.4 Relation between primal and dual

By the Duality Theorem and Lemma 1.7, a linear programme admits a solution if and only if its dual admits a solution. Moreover, according to Lemma 1.8, if a linear programme is unbounded, then its dual is not feasible. Reciprocally, if a linear programme admits no feasible solution, then its dual is unbounded. Finally, it is possible that both a linear programme and its dual have no feasible solution as shown by the following example.

$$\begin{array}{rrcl} \text{Maximize} & 2x_1 - x_2 & & \\ \text{Subject to:} & x_1 - x_2 & \leq & 1 \\ & -x_1 + x_2 & \leq & -2 \\ & x_1, x_2 & \geq & 0 \end{array}$$

Besides the fact it provides a certificate of optimality, the Duality Theorem has also a practical interest in the application of the Simplex Method. Indeed, the time-complexity of the Simplex Method mainly yields in the number of constraints of the considered linear programme. Hence, when dealing with a linear programme with few variables and many constraints, it will be more efficient to apply the Simplex Method on its dual.

Another interesting application of the Duality Theorem is that it is possible to compute an optimal solution for the dual problem from an optimal solution of the primal. Doing so gives an easy way to test the optimality of a solution. Indeed, if you have a feasible solution of some linear programme, then a solution of the dual problem can be derived (as explained below). Then the initial solution is optimal if and only if the solution obtained for the dual is feasible and leads to the same value.

More formally, the following theorems can be proved

Theorem 1.11 (Complementary Slackness)**.** *Let $(x_1, \dots, x_n)$ be a feasible solution of Problem 1.15 and $(y_1, \dots, y_m)$ be a feasible solution of Problem 1.16. These are optimal solutions if and only if*

$$\begin{array}{l} \sum_{i=1}^{m} a_{ij} y_i = c_j, \quad \text{or } x_j = 0, \quad \text{or both} \quad \text{for all } 1 \leq j \leq n, \text{ and} \\ \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad \text{or } y_i = 0, \quad \text{or both} \quad \text{for all } 1 \leq i \leq m. \end{array}$$

Proof. First, we note that since x and y are feasible $(b_i - \sum_{j=1}^{n} a_{ij}x_j)y_i \geq 0$ and $(\sum_{i=1}^{m} a_{ij}y_i - c_j)x_j \geq 0$. Summing these inequalities over i and j, we obtain

$$\sum_{i=1}^{m} \left(b_i - \sum_{j=1}^{n} a_{ij} x_j \right) y_i \quad \geq \quad 0 \tag{1.24}$$

$$\sum_{j=1}^{n} \left(\sum_{i=1}^{n} a_{ij} y_i - c_j \right) x_j \quad \geq \quad 0 \tag{1.25}$$

Adding Inequalities 1.24 and 1.25 and using the Duality Theorem (Theorem 1.14), we obtain

$$\sum_{i=1}^{m} b_i y_i - \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j y_i + \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} y_i x_j - \sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i - \sum_{j=1}^{n} c_j x_j = 0.$$

Therefore, Inequalities 1.24 and 1.25 must be equalities. As the variables are positive, we further get that

$$\begin{aligned} \text{for all } i, \qquad & \left(b_i - \sum_{j=1}^{n} a_{ij}x_j\right) y_i = 0 \\ \text{and for all } j, \qquad & \left(\sum_{i=1}^{m} a_{ij}y_i - c_j\right) x_j = 0. \end{aligned}$$

A product is equal to zero if one of its two members is null and we obtain the desired result. □

Theorem 1.12. *A feasible solution $(x_1,\dots,x_n)$ of Problem 1.15 is optimal if and only if there is a feasible solution $(y_1,\dots,y_m)$ of Problem 1.16 such that:*

$$\begin{array}{rcr} \sum_{i=}^{m} a_{ij}y_i = c_j & if & x_j > 0 \\ y_i = 0 & if & \sum_{j=1}^{m} a_{ij}x_j < b_i \end{array} \tag{1.26}$$

Note that, if Problem 1.15 admits a non-degenerate solution $(x_1,\dots,x_n)$, i.e., $x_i > 0$ for any $i \le n$, then the system of equations in Theorem 1.12 admits a unique solution.

Optimality certificates - Examples. Let see how to apply this theorem on two examples.

Let us first examine the statement that

$$x_1^* = 2,\ x_2^* = 4,\ x_3^* = 0,\ x_4^* = 0,\ x_5^* = 7,\ x_6^* = 0$$

is an optimal solution of the problem

$$\begin{array}{lrcrcrcrcrcrcl} \text{Maximize} & 18x_1 & - & 7x_2 & + & 12x_3 & + & 5x_4 & & & + & 8x_6 & & \\ \text{Subject to:} & 2x_1 & - & 6x_2 & + & 2x_3 & + & 7x_4 & + & 3x_5 & + & 8x_6 & \le & 1 \\ & -3x_1 & - & x_2 & + & 4x_3 & - & 3x_4 & + & x_5 & + & 2x_6 & \le & -2 \\ & 8x_1 & - & 3x_2 & + & 5x_3 & - & 2x_4 & & & + & 2x_6 & \le & 4 \\ & 4x_1 & & & + & 8x_3 & + & 7x_4 & - & x_5 & + & 3x_6 & \le & 1 \\ & 5x_1 & + & 2x_2 & - & 3x_3 & + & 6x_4 & - & 2x_5 & - & x_6 & \le & 5 \\ & & & & & & & & & \multicolumn{3}{r}{x_1,x_2,\cdots,x_6} & \ge & 0 \end{array}$$

In this case, (1.26) says:

$$\begin{array}{rcrcrcrcrcr} 2y_1^* & - & 3y_2^* & + & 8y_3^* & + & 4y_4^* & + & 5y_5^* & = & 18 \\ -6y_1^* & - & y_2^* & - & 3y_3^* & & & + & 2y_5^* & = & -7 \\ 3y_1^* & + & y_2^* & & & - & y_4^* & - & 2y_5^* & = & 0 \\ & & y_2^* & & & & & & & = & 0 \\ & & & & & & & & y_5^* & = & 0 \end{array}$$

As the solution $(\frac{1}{3},0,\frac{5}{3},1,0)$ is a feasible solution of the dual problem (Problem 1.16), the proposed solution is optimal.

Secondly, is

$$x_1^* = 0,\ x_2^* = 2,\ x_3^* = 0,\ x_4^* = 7,\ x_5^* = 0$$

an optimal solution of the following problem?

$$\begin{array}{lrcrcrcrcrcl}
\text{Maximize} & 8x_1 & - & 9x_2 & + & 12x_3 & + & 4x_4 & + & 11x_5 & & \\
\text{Subject to:} & 2x_1 & - & 3x_2 & + & 4x_3 & + & x_4 & + & 3x_5 & \leq & 1 \\
 & x_1 & + & 7x_2 & + & 3x_3 & - & 2x_4 & + & x_5 & \leq & 1 \\
 & 5x_1 & + & 4x_2 & - & 6x_3 & + & 2x_4 & + & 3x_5 & \leq & 22 \\
 & & & & & & & \multicolumn{3}{r}{x_1, x_2, \cdots, x_5} & \geq & 0
\end{array}$$

Here (1.26) translates into:

$$\begin{array}{rcrcrcr}
-3y_1^* & + & 7y_2^* & + & 4y_3^* & = & -9 \\
y_1^* & - & 2y_2^* & + & 2y_3^* & = & 4 \\
 & & y_2^* & & & = & 0
\end{array}$$

As the unique solution of the system $(3.4, 0, 0.3)$ is not a feasible solution of Problem 1.16, the proposed solution is not optimal.

1.3.5 Interpretation of dual variables

As said in the introduction of this section, one of the major interests of the dual programme is that, in some problems, the variables of the dual problem have an interpretation.

A classical example is the *economical interpretation* of the dual variables of the following problem. Consider the problem that consits in maximizing the benefit of a company building some products. Each variable x_j of the primal problem measures the amount of product j that is built, and b_i the amount of resource i (needed to build the products) that is available. Note that, for any $i \leq n$, $j \leq m$, $a_{i,j}$ represents the number of units of resource i needed per unit of product j. Finally, c_j denotes the benefit (the price) of a unit of product j.

Hence, by checking the units of measure in the constraints $\sum a_{ij} y_i \geq c_j$, the variable y_i must represent a benefit per unit of resource i. Somehow, the variable y_i measures the unitary value of the resource i. This is illustrated by the following theorem the proof of which is omitted.

Theorem 1.13. *If Problem 1.15 admits a non-degenerate optimal solution with value z^*, then there is $\varepsilon > 0$ such that, for any $|t_i| \leq \varepsilon$ $(i = 1, \ldots, m)$, the problem*

$$\begin{array}{lll}
\textit{Maximize} & \sum_{j=}^{n} c_j x_j & \\
\textit{Subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i + t_i & (i = 1, \ldots, m) \\
 & x_j \geq 0 & (j = 1, \ldots, n)
\end{array}$$

admits an optimal solution with value $z^* + \sum_{i=1}^{m} y_i^* t_i$, where $(y_1^*, \ldots, y_m^*)$ is the optimal solution of the dual of Problem 1.15.

Theorem 1.13 shows how small variations in the amount of available resources can affect the benefit of the company. For any unit of extra resource i, the benefit increases by y_i^*. Sometimes, y_i^* is called the *marginal cost* of the resource i.

In many networks design problems, a clever interpretation of dual variables may help to achieve more efficient linear programme or to understand the problem better.

1.4 Exercices

1.4.1 General modelling

Exercise 1. Which problem(s) among (1.27), (1.28) and (1.29) are under the standard form?

$$\begin{array}{lrcrcr} \text{Maximize} & 3x_1 & - & 5x_2 & & \\ \text{Subject to:} & 4x_1 & + & 5x_2 & \geq & 3 \\ & 6x_1 & - & 6x_2 & = & 7 \\ & x_1 & + & 8x_2 & \leq & 20 \\ & & & x_1, x_2 & \geq & 0 \end{array} \tag{1.27}$$

$$\begin{array}{lrcrcrcrcr} \text{Minimize} & 3x_1 & + & x_2 & + & 4x_3 & + & x_4 & & \\ \text{Subject to:} & 9x_1 & + & 2x_2 & + & 6x_3 & + & 5x_4 & \leq & 5 \\ & 8x_1 & + & 9x_2 & + & 7x_3 & + & 9x_4 & \leq & 2 \\ & & & & & & & x_1, x_2, x_3 & \geq & 0 \end{array} \tag{1.28}$$

$$\begin{array}{lrcrcr} \text{Maximize} & 8x_1 & - & 4x_2 & & \\ \text{Subject to:} & 3x_1 & + & x_2 & \leq & 7 \\ & 9x_1 & + & 5x_2 & \leq & -2 \\ & & & x_1, x_2 & \geq & 0 \end{array} \tag{1.29}$$

Exercise 2. Put under the standard form:

$$\begin{array}{lrcrcrcrcr} \text{Minimize} & -8x_1 & + & 9x_2 & + & 2x_3 & - & 6x_4 & & \\ \text{Subject to:} & 6x_1 & + & 6x_2 & - & 10x_3 & + & 2x_4 & \geq & 3 \\ & & & & & & & x_1, x_2, x_3, x_4 & \geq & 0 \end{array}$$

Exercise 3. a) Put under the standard form:

$$\begin{array}{lrcrcrcr} \text{Maximize} & -6x_1 & + & 4x_2 & + & 2x_3 & & \\ \text{Subject to:} & 4x_1 & + & 5x_2 & - & x_3 & = & 3 \\ & 2x_1 & + & 3x_2 & + & 2x_3 & \leq & 4 \\ & & & & & x_1, x_2, x_3 & \geq & 0 \end{array}$$

Could you reformulate the above linear programme using only 2 variables?

b) Write a linear programme with only non negative variables equivalent to the one below:

$$\begin{array}{lrcrcr} \text{Maximize} & 8x_1 & - & 4x_2 & & \\ \text{Subject to:} & 3x_1 & + & x_2 & \leq & 7 \\ & 9x_1 & + & 5x_2 & \leq & -2 \\ & & & x_1 & \geq & 0 \\ & & & x_2 & \in & \mathbb{R} \end{array}$$

Hint: You may have to add additional variables.

Exercise 4. Show that the linear programme (1.30) has no feasible solutions and that the linear programme (1.31) is unbounded.

$$\begin{array}{lrcrcr} \text{Maximize} & 3x_1 & - & x_2 & & \\ \text{Subject to:} & x_1 & + & x_2 & \leq & 2 \\ & -2x_1 & - & 2x_2 & \leq & -10 \\ & & & x_1, x_2 & \geq & 0 \end{array} \tag{1.30}$$

$$\begin{array}{lrcrcr} \text{Maximize} & x_1 & - & x_2 & & \\ \text{Subject to:} & -2x_1 & + & x_2 & \leq & -1 \\ & -x_1 & - & 2x_2 & \leq & -2 \\ & & & x_1, x_2 & \geq & 0 \end{array} \tag{1.31}$$

Exercise 5. Find necessary and sufficient conditions on the numbers s and t for the problem

$$\begin{array}{lrcrcr} \text{Maximize} & x_1 & + & x_2 & & \\ \text{Subject to:} & sx_1 & + & tx_2 & \leq & 1 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

a) to admit an optimal solution;

b) to be unfeasible;

c) to be unbounded.

Exercise 6. Prove or disprove: if the following linear programme (1.32)

$$\begin{array}{lrcll} \text{Maximize} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for all } 1 \leq i \leq m \\ & x_j & \geq & 0 & \text{for all } 1 \leq j \leq n. \end{array} \tag{1.32}$$

is *unbounded*, then there exists an index k such that the problem:

$$\begin{array}{lrcll} \text{Maximize} & x_k & & & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for } 1 \leq i \leq m \\ & x_j & \geq & 0 & \text{for } 1 \leq j \leq n \end{array}$$

is unbounded.

Exercise 7. The factory RadioIn builds to types of radios A and B. Every radio is produced by the work of three specialists Pierre, Paul and Jacques. Pierre works at most 24 hours per week. Paul works at most 45 hours per week. Jacques works at most 30 hours per week. The resources necessary to build each type of radio and their selling prices as well are given in the following table:

	Radio A	Radio B
Pierre	1h	2h
Paul	2h	1h
Jacques	1h	3h
Selling prices	15 euros	10 euros

We assume that the company has no problem to sell its production, whichever it is.

a) Model the problem of finding a weekly production plan maximizing the revenue of RadioIn as a linear programme. Write precisely what are the decision variables, the objective function and the constraints.

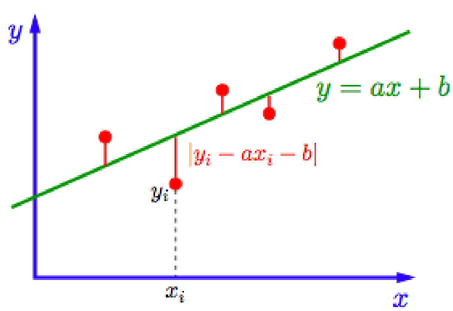
b) Solve the linear programme using the geometric method and give the optimal production plan.

Exercise 8. The following table shows the different possible schedule times for the drivers of a bus company. The company wants that at least one driver is present at every hour of the working day (from 9 to 17). The problem is to determine the schedule satisfying this condition with minimum cost.

Time	9 – 11h	9 – 13h	11 – 16h	12 – 15h	13 – 16h	14– 17h	16 – 17h
Cost	18	30	38	14	22	16	9

Formulate an integer linear programme that solves the company decision problem.

Exercise 9 (Chebyshev's approximation)**.** Data : m measures of points $(x_i, y_i) \in \mathbb{R}^2$, $i = 1, \dots, m$. Objective: Determine a linear approximation $y = ax + b$ minimizing the largest error of approximation. The decision variables of this problem are $a \in \mathbb{R}$ and $b \in \mathbb{R}$. The problem may be



formulated as:

$$\min z = \max_{i=1,\dots,m} \{|y_i - ax_i - b|\}.$$

It is unfortunately not under the form of a linear programme. Let us try to do some transformations.

Questions:

1. We call MIN-MAX the problem of minimizing the maximum of a set of numbers:

$$\min z \ \text{ with } \ z = \max\{c_1 x, \dots, c_k x\}.$$

How to write a MIN-MAX as a linear programme?

2. Can we express the following constraints

$$|x| \leq b$$

or

$$|x| \geq b$$

in a linear problem (that is without absolute values)? If yes, how?

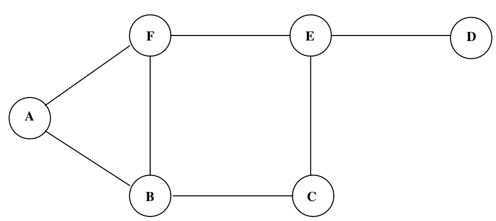
3. Rewrite the problem of finding a Chebyshev's linear approximation as a linear programme.

1.4.2 Modelling Combinatorial Problems via (integer) linear programming

Lots of combinatorial problems may be formulated as linear programmes.

Exercise 10 (MINIMUM VERTEX COVER)**.** A *vertex cover* in a graph $G = (V, E)$ is a set K of vertices such that each edge e of E is incident to at least one vertex of K. The MINIMUM VERTEX COVER problem is to find a vertex cover of minimum cardinality in a given graph.

1. Express MINIMUM VERTEX COVER for the *following graph* as an integer linear programme:



2. Express MINIMUM VERTEX COVER for a *general graph* as a linear programme.

General knowledge: The MINIMUM VERTEX COVER is an NP-complete problem. However, there exist approximation algorithms to solve it with a factor of approximation 2.

Exercise 11 (MINIMUM EDGE COVER)**.** An *edge cover* of a graph $G = (V, E)$ is a set of edges $F \subseteq E$ such that every vertex $v \in V$ is incident to at least one edge of F. The MINIMUM EDGE COVER problem is to find an edge cover of minimum cardinality in a given graph.

Adapt the integer linear programme modelling MINIMUM VERTEX COVER to obtain an integer linear programming formulation of MINIMUM EDGE COVER.

Exercise 12. Consider the graph

```
A
|
B–D
| /
C
```

What does the following linear programme do?

$$
\begin{array}{rc}
\text{Minimize} & x_A + x_B + x_C + x_D \\
\text{Subject to:} & \\
& x_A + x_B \geq 1 \\
& x_B + x_D \geq 1 \\
& x_B + x_C \geq 1 \\
& x_C + x_D \geq 1 \\
& x_A \geq 0, x_B \geq 0, x_C \geq 0, x_D \geq 0
\end{array}
$$

Exercise 13 (Maximum cardinality matching problem (Polynomial < flows or augmenting paths))**.** Let $G = (V,E)$ be a graph. Recall that a *matching* $M \subseteq E$ is a set of edges such that every vertex of V is incident to at most one edge of M. The MAXIMUM MATCHING problem is to find a matching M of maximum size. Express MAXIMUM MATCHING as a integer linear programme.

General knowledge: MAXIMUM MATCHING is polynomial problem.

Exercise 14 (MAXIMUM CLIQUE)**.** Recall that a *clique* of a graph $G = (V,E)$ is a subset C of V, such that every two vertices in V are joined by an edge of E. The MAXIMUM CLIQUE problem consist of finding the largest cardinality of a clique.

Express MAXIMUM CLIQUE as an integer linear programme.

General knowledge: MAXIMUM CLIQUE is an NP-complete problem.

Exercise 15 (French newspaper enigma)**.** What is the maximum size of a set of integers between 1 and 100 such that for any pair (a,b), the difference a-b is not a square ?

1. Model this problem as a graph problem.
2. Write a linear programme to solve it.

Exercise 16 (Resource assignment)**.** A university class has to go from Marseille to Paris using buses. There are some strong inimities inside the group and two people that dislike each other cannot share the same bus. What is the minimum number of buses needed to transport the whole group? Write a LP that solve the problem. (We suppose that a bus does not have a limitation on the number of places.)

Exercise 17 (MAXIMUM STABLE SET)**.** Recall that a *stable set* of a graph $G = (V,E)$ is a subset S of pairwise non-adjacent vertices. The MAXIMUM STABLE SET problem consist in finding the largest cardinality of a stable set. Give a linear programming formulation of this problem.

Exercise 18 (MINIMUM SET COVER)**.** Let $U = \{1,...,n\}$ be a set and $\mathcal{S} = \{S_1, \ldots, S_m\}$ a set of subsets of $\mathcal{U}$. An *$\mathcal{S}$-cover* of U is a subset of $\mathcal{T}$ of $\mathcal{S}$ such that $\bigcup_{T\in\mathcal{T}} T = \mathcal{U}$. The MINIMUM SET COVER problem consists in, given a set U $\mathcal{S}$, finding an $\mathcal{S}$-cover of U of minimum cardinality.
Example: $U = \{1,2,3,4,5,6\}$, $S_1 = \{1,3,6\}$, $S_2 = \{2,4\}$, $S_3 = \{4,6\}$, $S_4 = \{1\}$, $S_5 = \{2,4,5\}$. $C = \{S_1,S_2,S_3\}$ is a set cover of cardinality 3. $C^* = \{S_1,S_5\}$ is a minimum set cover of cardinality 2.

Formulate MINIMUM SET COVER as an integer linear programme.

General knowledge: The associate decision problem k-SET COVER, which consists in deciding wether U has an $\mathcal{S}$-cover of cardinality at most k is $\mathcal{NP}$-complete.

Exercise 19 (Instance of MAXIMUM SET PACKING)**.** Suppose you are at a convention of foreign ambassadors, each of which speaks English and other various languages.

- French ambassador: French, Russian
- US ambassador:
- Brazilian ambassador: Portuguese, Spanish
- Chinese ambassador: Chinese, Russian
- Senegalese ambassador: Wolof, French, Spanish

You want to make an announcement to a group of them, but because you do not trust them, you do not want them to be able to speak among themselves without you being able to understand them (you only speak English). To ensure this, you will choose a group such that no two ambassadors speak the same language, other than English. On the other hand you also want to give your announcement to as many ambassadors as possible.

Write a linear programme giving the maximum number of ambassadors at which you will be able to give the message.

Exercise 20 (MAXIMUM SET PACKING)**.** Given a finite set S and a list $\mathcal{L}$ of subsets of S.
The MAXIMUM SET PACKING problem consists in finding the maximum number of pairwise disjoint sets in a given list $\mathcal{L}$. Give a linear programming formulation of this problem.

1.4.3 Modelling flows and shortest paths.

Recall that an *elementary flow network* is a four-tuple $N = (D,s,t,c)$ where

- $D = (V,A)$.
- c is a capacity function from A to $\mathbb{R}^+ \cup \infty$. For an arc $a \in A$, $c(a)$ represents its capacity, that is the maximum amount of flow it can carry.
- s and t are two distinct vertices: s is the source of the flow and t the sink. In an *elementary network*, the source has no incoming arcs and the destination has no outgoing arcs.

A *flow* is a function f from A to $\mathbb{R}^+$ which respects the flow conservation constraints and the capacity constraints.

General knowledge: MAXIMUM FLOW is a polynomial problem. It can be solved using the Floyd Fulkerson algorithm.

Exercise 21 (MAXIMUM FLOW)**.** Write a linear programming formulation of the MAXIMUM FLOW problem.

Exercise 22 (MULTICOMMODITY FLOW)**.** Consider a flow network $\mathcal{N} = (D,s,t,c)$. Consider a set of k commodities (s^c,t^c,d^c) with $1 \leq c \leq k$, where where d^c is the amount of flow that has to be sent from node s^c to node t^c. The multicommodity flow problem is to determine if all demands can be simultaneously routed on the network. This problem models a telecom network and is one of the fundamental problem of the networking research field.

Write a linear program that solves the multicommodity flow problem.

General knowledge: MULTICOMMODITY FLOW is an NP-complete problem as soon as the number of commodity is larger or equal to 2.

Exercise 23 (SHORTEST (s,t)-PATH)**.** Let $D = (V,A,l)$ be a an arc-weighted digraph with l is a length function from A to $\mathbb{R}^+$. For $a \in A$, l(a) is the length of arc a. Let s and t two distinguished vertices.

Write a linear programme that finds the length of a shortest path between s and t.

General knowledge: SHORTEST (s,t)-PATH is a polynomial problem. It can be solved using the Dijskstra algorithm.

Exercise 24 (Eccentricy and diameter)**.** The *distance* between two vertices in a graph is the number of edges in a shortest path connecting them. The *eccentricity* ε of a vertex v is the greatest distance between v and any other vertex. It can be thought of as how far a node is from the node most distant from it in the graph. The *diameter* of a graph is the maximum eccentricity of any vertex in the graph. That is, it is the greatest distance between any pair of vertices.

1. Write a linear programme to compute the eccentricity of a given vertex.

2. Write a linear programme which computes the diameter of a graph.

Exercise 25 (MINIMUM (s,t)-CUT)**.** Recall that in a flow network $N = (G,s,t,c)$ an (s,t)-*cut* is a bipartition $C = (V_s,V_t)$ of the vertices of G such that $s \in V_s$ and $t \in V_t$. The capacity of the cut C, denoted by $\delta(C)$, is the sum of the capacities of the out-arcs of V_s (i.e., the arcs (u,v) with $u \in V_s$ and $v \in V_t$).

Write a linear programme that finds the minimum capacity of an (s,t)-cut.

Hint: Use variables to know in which partition is each vertex and additional variables to know which edges are in the cut.

1.4.4 Simplex

Exercise 26. Solve with the Simplex Method the following problems:

a.

$$\begin{array}{lrcrcrcl}
\text{Maximize} & 3x_1 & + & 3x_2 & + & 4x_3 & & \\
\text{Subject to:} & & & & & & & \\
& x_1 & + & x_2 & + & 2x_3 & \leq & 4 \\
& 2x_1 & & & + & 3x_3 & \leq & 5 \\
& 2x_1 & + & x_2 & + & 3x_3 & \leq & 7 \\
& & & & & x_1, x_2, x_3 & \geq & 0
\end{array}$$

b.

$$\begin{array}{lrcrcrcrcl}
\text{Maximize} & 5x_1 & + & 6x_2 & + & 9x_3 & + & 8x_4 & & \\
\text{Subject to:} & & & & & & & & & \\
& x_1 & + & 2x_2 & + & 3x_3 & + & x_4 & \leq & 5 \\
& x_1 & + & x_2 & + & 2x_3 & + & 3x_4 & \leq & 3 \\
& & & & & & & x_1, x_2, x_3, x_4 & \geq & 0
\end{array}$$

c.

$$\begin{array}{lrcrcl}
\text{Maximize} & 2x_1 & + & x_2 & & \\
\text{Subject to:} & & & & & \\
& 2x_1 & + & 3x_2 & \leq & 3 \\
& x_1 & + & 5x_2 & \leq & 1 \\
& 2x_1 & + & x_2 & \leq & 4 \\
& 4x_1 & + & x_2 & \leq & 5 \\
& & & x_1, x_2 & \geq & 0
\end{array}$$

Exercise 27. Use the Simplex Method to describe *all* the optimal solutions of the following linear programme:

$$\begin{array}{lrcrcrcrcl}
\text{Maximize} & 2x_1 & + & 3x_2 & + & 5x_3 & + & 4x_4 & & \\
\text{Subject to:} & & & & & & & & & \\
& x_1 & + & 2x_2 & + & 3x_3 & + & x_4 & \leq & 5 \\
& x_1 & + & x_2 & + & 2x_3 & + & 3x_4 & \leq & 3 \\
& & & & & & & x_1, x_2, x_3, x_4 & \geq & 0
\end{array}$$

Exercise 28. Solve the following problems using the Simplex Method in two phases.

a.

$$\begin{array}{lrcrcr}
\text{Maximise} & 3x_1 & + & x_2 & & \\
\text{Subject to:} & & & & & \\
& x_1 & - & x_2 & \leq & -1 \\
& -x_1 & - & x_2 & \leq & -3 \\
& 2x_1 & + & x_2 & \leq & 4 \\
& & & x_1, x_2 & \geq & 0
\end{array}$$

b.

$$\begin{array}{lrcrcr}
\text{Maximise} & 3x_1 & + & x_2 & & \\
\text{Subject to:} & & & & & \\
& x_1 & - & x_2 & \leq & -1 \\
& -x_1 & - & x_2 & \leq & -3 \\
& 2x_1 & + & x_2 & \leq & 2 \\
& & & x_1, x_2 & \geq & 0
\end{array}$$

c.

$$\begin{array}{lrcrcr}
\text{Maximise} & 3x_1 & + & x_2 & & \\
\text{Subject to:} & & & & & \\
& x_1 & - & x_2 & \leq & -1 \\
& -x_1 & - & x_2 & \leq & -3 \\
& 2x_1 & - & x_2 & \leq & 2 \\
& & & x_1, x_2 & \geq & 0
\end{array}$$

1.4.5 Duality

Exercise 29. Write the dual of the following linear programme.

$$\begin{array}{lrcrcr}
\text{Maximize} & 7x_1 & + & x_2 & & \\
\text{Subject to:} & & & & & \\
& 4x_1 & + & 3x_2 & \leq & 3 \\
& x_1 & - & 2x_2 & \leq & 4 \\
& -5x_1 & - & 2x_2 & \leq & 3 \\
& & & x_1, x_2 & \geq & 0
\end{array}$$

Exercise 30. Consider the following linear programme.

$$\begin{array}{lrcrcrcrcr}
\text{Minimize} & -2x_1 & - & 3x_2 & - & 2x_3 & - & 3x_4 & & \\
\text{Subject to:} & & & & & & & & & \\
& -2x_1 & - & x_2 & - & 3x_3 & - & 2x_4 & \geq & -8 \\
& 3x_1 & + & 2x_2 & + & 2x_3 & + & x_4 & \leq & 7 \\
& & & & & & x_1, x_2, & x_3, x_4 & \geq & 0
\end{array} \tag{1.33}$$

a) Write the programme (1.40) under the standard form.

b) Write the dual (D) of programme (1.40).

c) Give a graphical solution of the dual programme (D).

d) Carry on the first iteration of the Simplex Method on the linear programme (1.40).

   After three iterations, one find that the optimal solution of this programme is $x_1 = 0$, $x_2 = 2$, $x_3 = 0$ and $x_4 = 3$.

e) Verify that the solution of (D) obtained at Question c) is optimal.

Exercise 31. Prove that the following linear programme is unbounded.

$$\begin{array}{llrcrcrcr}
\text{Maximize} & 3x_1 & - & 4x_2 & + & 3x_3 & & \\
\text{Subject to :} & & & & & & & \\
& -x_1 & + & x_2 & + & x_3 & \leq & -3 \\
& -2x_1 & - & 3x_2 & + & 4x_3 & \leq & -5 \\
& -3x_1 & + & 2x_2 & - & x_3 & \leq & -3 \\
& & & & & x_1,x_2,x_3 & \geq & 0
\end{array}$$

Exercise 32. We consider the following linear programme.

$$\begin{array}{llrcrcrcr}
\text{Maximize} & x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{Subject to :} & & & & & & & \\
& 2x_1 & - & x_2 & + & x_3 & \leq & 4 \\
& -4x_1 & + & 3x_2 & & & \leq & 2 \\
& 3x_1 & - & 2x_2 & - & x_3 & \leq & 5 \\
& & & & & x_1,x_2,x_3 & \geq & 0
\end{array}$$

Is the solution $x_1^*=0$, $x_2^*=0$, $x_3^*=4$ optimal?

Exercise 33. We consider the following linear programme.

$$\begin{array}{llrcrcrcrcrcr}
\text{Maximize} & 7x_1 & + & 6x_2 & + & 5x_3 & - & 2x_4 & + & 3x_5 & & \\
\text{Subject to:} & & & & & & & & & & & \\
& x_1 & + & 3x_2 & + & 5x_3 & - & 2x_4 & + & 2x_5 & \leq & 4 \\
& 4x_1 & + & 2x_2 & - & 2x_3 & + & x_4 & + & x_5 & \leq & 3 \\
& 2x_1 & + & 4x_2 & + & 4x_3 & - & 2x_4 & + & 5x_5 & \leq & 5 \\
& 3x_1 & + & x_2 & + & 2x_3 & - & x_4 & - & 2x_5 & \leq & 1 \\
& & & & & & & x_1,x_2,x_3,x_4,x_5 & & & \geq & 0.
\end{array}$$

Is the solution $x_1^*=0, x_2^*=\frac{4}{3}, x_3^*=\frac{2}{3}, x_4^*=\frac{5}{3}, x_5^*=0$, optimal?

Exercise 34. 1. Because of the arrival of new models, a salesman wants to sell off quickly its stock composed of eight phones, four hands-free kits and nineteen prepaid cards. Thanks to a market study, he knows that he can propose an offer with a phone and two prepaid cards and that this offer will bring in a profit of seven euros. Similarly, we can prepare a box with a phone, a hands-free kit and three prepaid cards, yielding a profit of nine euros. He is assured to be able to sell any quantity of these two offers within the availability of its stock. What quantity of each offer should the salesman prepare to maximize its net profit?

2. A sales representative of a supermarket chain proposes to buy its stock (the products, not the offers). What unit prices should he negociate for each product (phone, hands-free kits, and prepaid cards)?

Exercise 35 (FARKAS' LEMMA)**.** The following two linear programmes are duals of each other.

$$\begin{array}{lrlcl}
\text{Maximize} & \mathbf{0^T x} & \text{subject to} & \mathbf{Ax}=\mathbf{0} & \text{and } \mathbf{x}\geq\mathbf{b} \\
\text{Minimize} & -\mathbf{b^T z} & \text{subject to} & \mathbf{A^T y}-\mathbf{z}=\mathbf{0} & \text{and } \mathbf{z}\geq\mathbf{0}
\end{array}$$

Farkas' Lemma says that exactly one of the two linear systems:

$$\mathbf{Ax} = \mathbf{0}, \;\; \mathbf{x} \geq \mathbf{b} \quad \text{and} \quad \mathbf{yA} \geq \mathbf{0}, \;\; \mathbf{yAb} > 0$$

has a solution. Deduce Farkas' Lemma from the Duality Theorem (Theorem 1.14 below).

We recall the duality theorem:

Theorem 1.14 (DUALITY THEOREM)**.** *If the primal problem defined by Problem 1.15 admits an optimal solution $(x_1^*, \ldots, x_n^*)$, then the dual problem (Problem 1.16) admits an optimal solution $(y_1^*, \ldots, y_m^*)$, and*

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*.$$

with

Problem 1.15.

$$\begin{array}{rll} \text{Maximize} & \sum_{j=1}^{n} c_j x_j & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & \text{for all } \; 1 \leq i \leq m \\ & x_j \geq 0 & \text{for all } 1 \leq j \leq n \end{array}$$

and

Problem 1.16.

$$\begin{array}{rll} \text{Minimize} & \sum_{i=1}^{m} b_i y_i & \\ \text{Subject to:} & \sum_{i=1}^{m} a_{ij} y_i \geq c_j & \text{for all } \; 1 \leq j \leq n \\ & y_i \geq 0 & \text{for all } \; 1 \leq i \leq m \end{array}$$

Exercise 36. The following two linear programmes are duals of each other.

$$\begin{array}{rlll} \text{Minimize } \mathbf{0}^T\mathbf{y} & \text{subject to} & \mathbf{A}^T\mathbf{y} \geq \mathbf{c} & \\ \text{Maximize } \mathbf{c}^\mathbf{T}\mathbf{x} & \text{subject to} & \mathbf{Ax} = \mathbf{0} & \text{and } \; \mathbf{x} \geq \mathbf{0} \end{array}$$

A variant of Farkas' Lemma says that exactly one of the two linear systems:

$$\mathbf{A}^\mathbf{T}\mathbf{y} \geq \mathbf{c} \quad \text{and} \quad \mathbf{Ax} = \mathbf{0}, \;\; \mathbf{x} \geq \mathbf{0}, \;\; \mathbf{cx} > 0$$

has a solution. Deduce this variant of Farkas' Lemma from the Duality Theorem (Theorem 1.14 above).

Exercise 37 (Application of duality to game theory- Minimax principle)**.** In this problem, based on a lecture of Shuchi Chawla, we present an application of linear programming duality in the theory of games. In particular, we will prove the Minimax Theorem using duality.

Let us first give some definition. A *two-players zero-sum game* is a protocol defined as follows: two players choose strategies in turn; given two strategies x and y, we have a *valuation function* $f(x,y)$ which tells us what the payoff for Player one is. Since it is a zero sum game, the payoff for the Player two is exactly $-f(x,y)$. We can view such a game as a matrix of payoffs for one of the players. As an example take the game of Rock-Paper-Scissors, where the payoff is one for the winning party or 0 if there is a tie. The matrix of winnings for player one will then be the following:

$$A = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$$

Where A_{ij} corresponds to the payoff for player one if player one picks the i-th element and player two the j-th element of the sequence (Rock, Paper, Scissors). We will henceforth refer to player number two as the column player and player number one as the row player. If the row player goes first, he obviously wants to minimize the possible gain of the column player.

What is the payoff of the row player? If the row player plays first, he knows that the column player will choose the minimum of the line he will choose. So he has to choose the line with the maximal minimum value. That is its payoff is

$$\max_i \min_j A_{ij}.$$

Similarly, what is the payoff of the column player if he plays first? If the column player plays first, the column player knows that the row player will choose the maximum of the column that will be chosen. So the column player has to choose the column with minimal maximum value. Hence, the payoff of the row player in this case is

$$\min_j \max_i A_{ij}.$$

Compare the payoffs. It is clear that

$$\max_i \min_j A_{ij} \leq \min_j \max_i A_{ij}.$$

The minimax theorem states that if we allow the players to choose probability distributions instead of a given column or row, then the payoff is the same no matter which player starts. More formally:

Theorem 1.17 (Minimax theorem)**.** *If x and y are probability vectors, then*

$$\max_{\mathbf{y}}(\min_{\mathbf{x}} \mathbf{y}^T \mathbf{A}\mathbf{x}) = \min_{\mathbf{x}}(\max_{\mathbf{y}}(\mathbf{y}^T \mathbf{A}\mathbf{x})).$$

Let us prove the theorem.

1. Formulate the problem of maximizing its payoff as a linear programme.
2. Formulate the second problem of minimizing its loss as a linear programme.
3. Prove that the second problem is a dual of the first problem.
4. Conclude.

Exercise 38. Prove the following proposition.

Proposition 1.18. *The dual problem of the problem*

$$\textit{Maximize } \mathbf{c}^T\mathbf{x} \ \textit{ subject to } \ \mathbf{A}\mathbf{x} \leq \mathbf{a} \ \textit{ and } \ \mathbf{B}\mathbf{x} = \mathbf{b} \ \textit{ and } \ \mathbf{x} \geq \mathbf{0}$$

is the problem

$$\textit{Minimize } \mathbf{a}^T\mathbf{y} + \mathbf{b}^T\mathbf{z} \ \textit{ subject to } \ \mathbf{A}^T\mathbf{y} + \mathbf{B}^T\mathbf{z} \geq \mathbf{c} \ \textit{ and } \ \mathbf{y} \geq \mathbf{0}.$$

Bibliography

[1] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley-Interscience, 1998.

[2] V. Chvátal. *Linear Programming*. W. H. Freeman and Company, New York, 1983.

[3] *ILOG CPLEX optimization software*. `http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/`.

[4] *GNU Linear Programming Kit*. `http://www.gnu.org/software/glpk/`.

[5] D. Gale, H. W. Kuhn, and A. W. Tucker. Linear Programming and the Theory of Games. In *T. C. Koopmans (ed.), Activity Analysis of Production and Allocation*, New York, Wiley, 1951.

[6] M. Sakarovitch. *Optimisation Combinatoire: Graphes et Programmation Linéaire*. Hermann, Paris, 1984.

Linear Programming: Introduction

Frédéric Giroire

Course Schedule

- Session 1: Introduction to optimization.
  Modelling and Solving simple problems.
  Modelling combinatorial problems.
- Session 2: Duality or Assessing the quality of a solution.
- Session 3: Solving problems in practice or using solvers (Glpk or Cplex).

`http://www-sop.inria.fr/members/Frederic.Giroire/teaching/ubinet/`

Motivation

Why linear programming is a very important topic?

- A lot of problems can be formulated as linear programmes, and
- There exist efficient methods to solve them
- or at least give good approximations.
- Solve difficult problems: e.g. original example given by the inventor of the theory, Dantzig. Best assignment of 70 people to 70 tasks.

$\rightarrow$ Magic algorithmic box.

What is a linear programme?

- Optimization problem consisting in
  - maximizing (or minimizing) a linear objective function
  - of n decision variables
  - subject to a set of constraints expressed by linear equations or inequalities.
- Originally, military context: "programme"="resource planning". Now "programme"="problem"
- Terminology due to George B. Dantzig, inventor of the Simplex Algorithm (1947)

Terminology

x_1, x_2 : Decision variables

$$\begin{array}{lll} \text{max} & 350x_1 + 300x_2 & \text{Objective function} \\ \text{subject to} & & \\ & x_1 + x_2 \leq 200 & \text{Constraints} \\ & 9x_1 + 6x_2 \leq 1566 & \\ & 12x_1 + 16x_2 \leq 2880 & \\ & x_1, x_2 \geq 0 & \end{array}$$

Terminology

x_1, x_2 : Decision variables

$$\begin{array}{lll} \text{max} & 350x_1 + 300x_2 & \text{Objective function} \\ \text{subject to} & & \\ & x_1 + x_2 \leq 200 & \text{Constraints} \\ & 9x_1 + 6x_2 \leq 1566 & \\ & 12x_1 + 16x_2 \leq 2880 & \\ & x_1, x_2 \geq 0 & \end{array}$$

In linear programme: objective function + constraints are all linear

Typically (not always): variables are non-negative

If variables are integer: system called Integer Programme (IP)

Terminology

Linear programmes can be written under the standard form:

$$\begin{array}{lrcll} \text{Maximize} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for all } 1 \leq i \leq m \\ & x_j & \geq & 0 & \text{for all } 1 \leq j \leq n. \end{array} \qquad (1)$$

- the problem is a maximization;
- all constraints are inequalities (and not equations);
- all variables are non-negative.

Example 1: a resource allocation problem

A company produces copper cable of 5 and 10 mm of diameter on a single production line with the following constraints:

- The available copper allows to produces 21 meters of cable of 5 mm diameter per week.
- A meter of 10 mm diameter copper consumes 4 times more copper than a meter of 5 mm diameter copper.

Due to demand, the weekly production of 5 mm cable is limited to 15 meters and the production of 10 mm cable should not exceed 40% of the total production. Cable are respectively sold 50 and 200 euros the meter.

What should the company produce in order to maximize its weekly revenue?

Example 1: a resource allocation problem

A company produces copper cable of 5 and 10 mm of diameter on a single production line with the following constraints:

- The available copper allows to produces 21 meters of cable of 5 mm diameter per week.
- A meter of 10 mm diameter copper consumes 4 times more copper than a meter of 5 mm diameter copper.

Due to demand, the weekly production of 5 mm cable is limited to 15 meters and the production of 10 mm cable should not exceed 40% of the total production. Cable are respectively sold 50 and 200 euros the meter.

What should the company produce in order to maximize its weekly revenue?

Example 1: a resource allocation problem

Define two decision variables:

- x_1: the number of thousands of meters of 5 mm cables produced every week
- x_2: the number of thousands meters of 10 mm cables produced every week

The revenue associated to a production (x_1, x_2) is

$$z = 50x_1 + 200x_2.$$

The capacity of production cannot be exceeded

$$x_1 + 4x_2 \leq 21.$$

Example 1: a resource allocation problem

The demand constraints have to be satisfied

$$x_2 \leq \frac{4}{10}(x_1 + x_2)$$

$$x_1 \leq 15$$

Negative quantities cannot be produced

$$x1 \geq 0, x2 \geq 0.$$

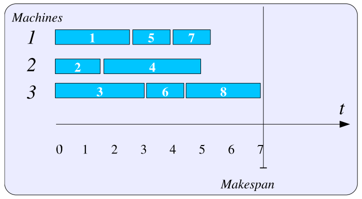
Example 1: a resource allocation problem

The model: To maximize the sell revenue, determine the solutions of the following linear programme x_1 and x_2:

$$\begin{array}{rc} \max & z = 50x_1 + 200x_2 \\ \text{subject to} & \\ & x_1 + 4x_2 \leq 21 \\ & -4x_1 + 6x_2 \leq 0 \\ & x_1 \leq 15 \\ & x_1, x_2 \geq 0 \end{array}$$

Example 2: Scheduling

- $m = 3$ machines
- $n = 8$ tasks
- Each task lasts x units of time

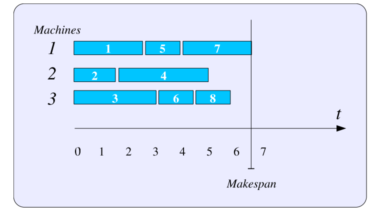


Objective: affect the tasks to the machines in order to minimize the duration

- Here, the 8 tasks are finished after 7 units of times on 3 machines.

Example 2: Scheduling

- $m = 3$ machines
- $n = 8$ tasks
- Each task lasts x units of time

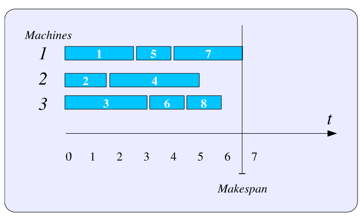


Objective: affect the tasks to the machines in order to minimize the duration

- Now, the 8 tasks are accomplished after 6.5 units of time: OPT?
- m^n possibilities! (Here $3^8 = 6561$)

Example 2: Scheduling

- $m = 3$ machines
- $n = 8$ tasks
- Each task lasts x units of time



Solution: LP model.

$$\begin{array}{lll} \min & t & \\ \text{subject to} & & \\ & \sum_{1 \le i \le n} t_i x_i^j \le t & (\forall j, 1 \le j \le m) \\ & \sum_{1 \le j \le m} x_i^j = 1 & (\forall i, 1 \le i \le n) \end{array}$$

with $x_i^j = 1$ if task i is affected to machine j.

Solving Difficult Problems

- Difficulty: Large number of solutions.
  - Choose the best solution among 2^n or $n!$ possibilities: all solutions cannot be enumerated.
  - Complexity of studied problems: often NP-complete.
- Solving methods:
  - Optimal solutions:
    - Graphical method (2 variables only).
    - Simplex method.
  - Approximations:
    - Theory of duality (assert the quality of a solution).
    - Approximation algorithms.

Graphical Method

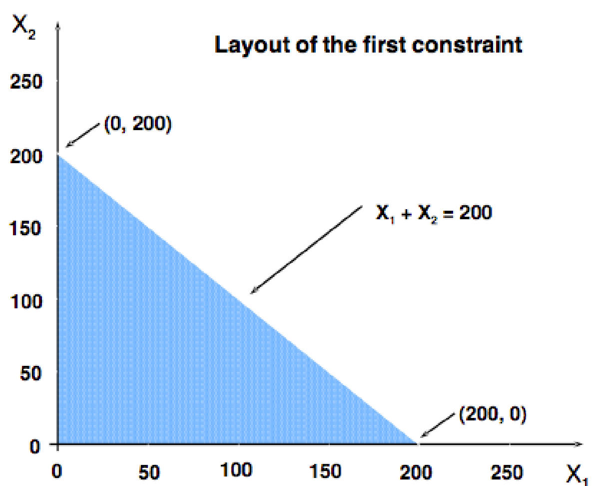
- The constraints of a linear programme define a zone of solutions.
- The best point of the zone corresponds to the optimal solution.
- For problem with 2 variables, easy to draw the zone of solutions and to find the optimal solution graphically.

Graphical Method

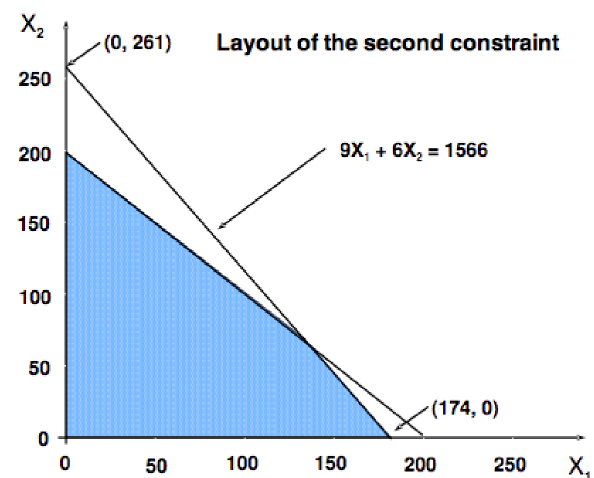
Example:

$$
\begin{array}{rl}
\max & 350x_1 + 300x_2 \\
\text{subject to} & \\
 & x_1 + x_2 \leq 200 \\
 & 9x_1 + 6x_2 \leq 1566 \\
 & 12x_1 + 16x_2 \leq 2880 \\
 & x_1, x_2 \geq 0
\end{array}
$$

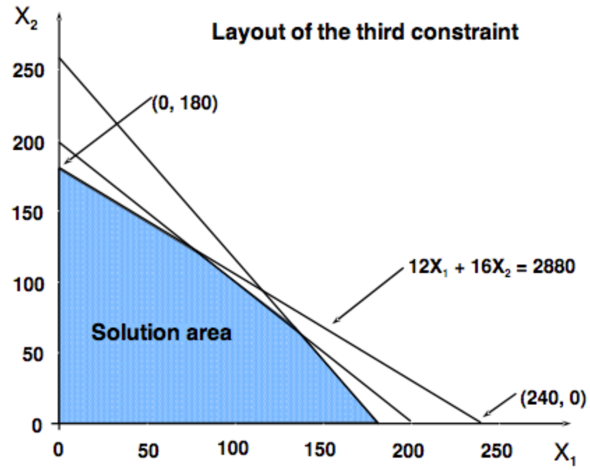
Graphical Method



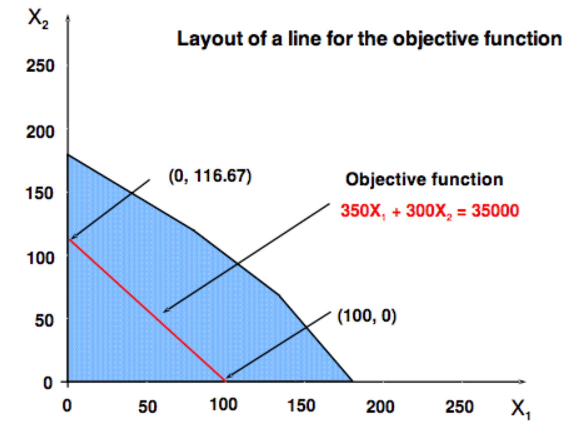
Graphical Method



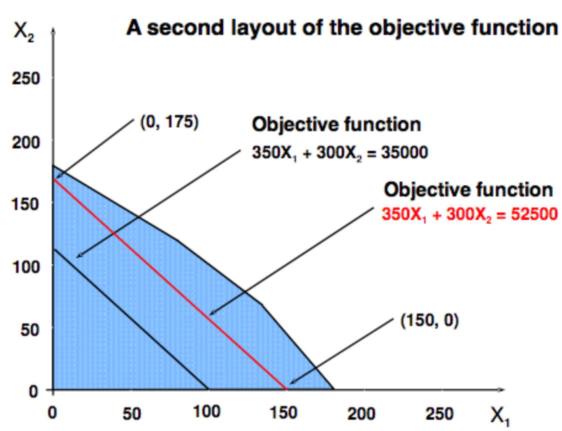
Graphical Method



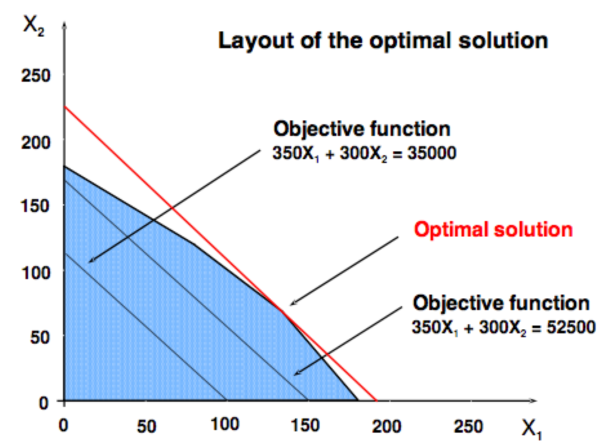
Graphical Method



Graphical Method



Graphical Method



Computation of the optimal solution

The optimal solution is at the intersection of the constraints:

$$x_1 + x_2 = 200 \quad (2)$$

$$9x_1 + 6x_2 = 1566 \quad (3)$$

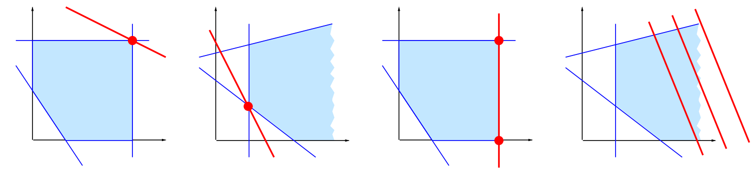
We get:

$$x_1 = 122$$

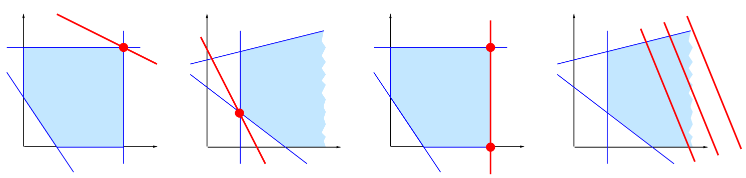
$$x_2 = 78$$

$$\text{Objective} = 66100.$$

Optimal Solutions: Different Cases



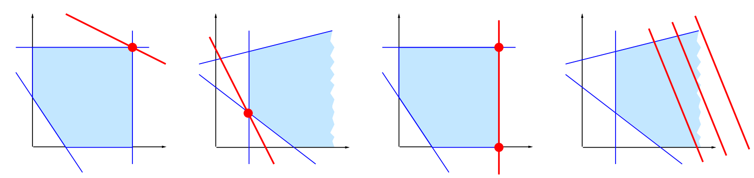
Optimal Solutions: Different Cases



Three different possible cases:

- a single optimal solution,
- an infinite number of optimal solutions, or
- no optimal solutions.

Optimal Solutions: Different Cases

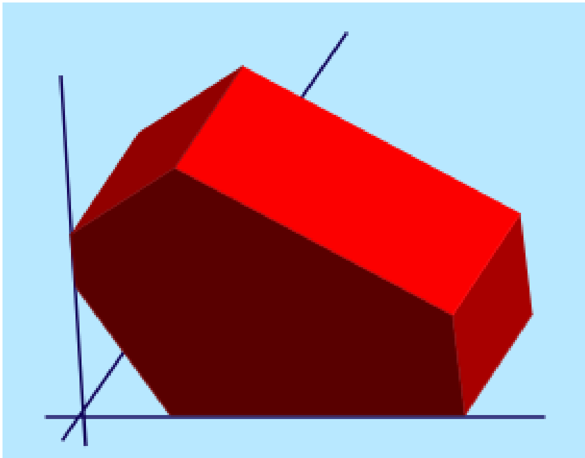


Three different possible cases:

- a single optimal solution,
- an infinite number of optimal solutions, or
- no optimal solutions.

If an optimal solution exists, there is always a corner point optimal solution!

Solving Linear Programmes



Solving Linear Programmes

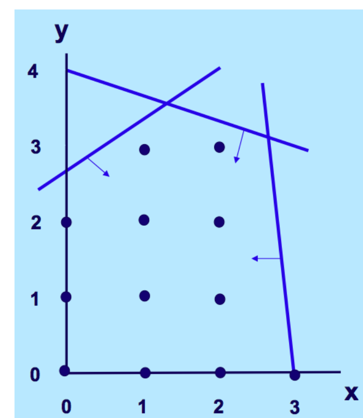
- The constraints of an LP give rise to a geometrical shape: a polyhedron.
- If we can determine all the corner points of the polyhedron, then we calculate the objective function at these points and take the best one as our optimal solution.
- The Simplex Method intelligently moves from corner to corner until it can prove that it has found the optimal solution.

Solving Linear Programmes

- Geometric method impossible in higher dimensions
- Algebraical methods:
  - Simplex method (George B. Dantzig 1949): skim through the feasible solution polytope.
    Similar to a "Gaussian elimination".
    Very good in practice, but can take an exponential time.
  - Polynomial methods exist:
    - Leonid Khachiyan 1979: ellipsoid method. But more theoretical than practical.
    - Narendra Karmarkar 1984: a new interior method. Can be used in practice.

But Integer Programming (IP) is different!

- Feasible region: a set of discrete points.
- Corner point solution not assured.
- No "efficient" way to solve an IP.
- Solving it as an LP provides a relaxation and a bound on the solution.



Summary: To be remembered

- What is a linear programme.
- The graphical method of resolution.
- Linear programs can be solved efficiently (polynomial).
- Integer programs are a lot harder (in general no polynomial algorithms).
  In this case, we look for approximate solutions.

Modelling Graph Problems using Linear Programmes: An Example

Frédéric Giroire

Maximum Cardinality Matching Problem

Definition. Let $G = (V, E)$ be a graph.

- A matching $M \subseteq E$ is a collection of edges such that every vertex of V is incident to at most one edge of M.
- The maximum cardinality matching problem is to find a matching M of maximum size.

Reminder: The problem is polynomial

- for a bipartite graph (augmenting paths or applications of flows)
- for a general graph (Edmund's algorithm)

Maximum Cardinality Matching Problem

An example:

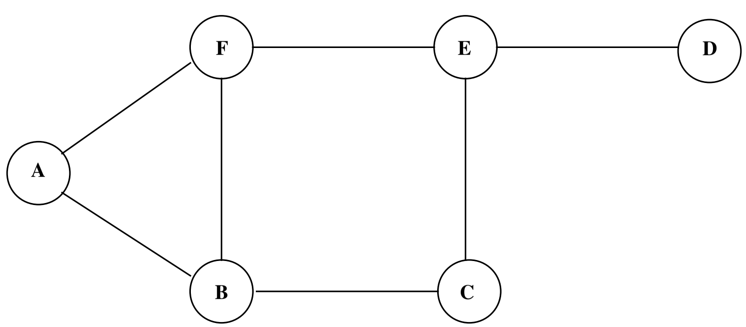


Figure: A graph with 6 vertices

Maximum Cardinality Matching Problem

An example:

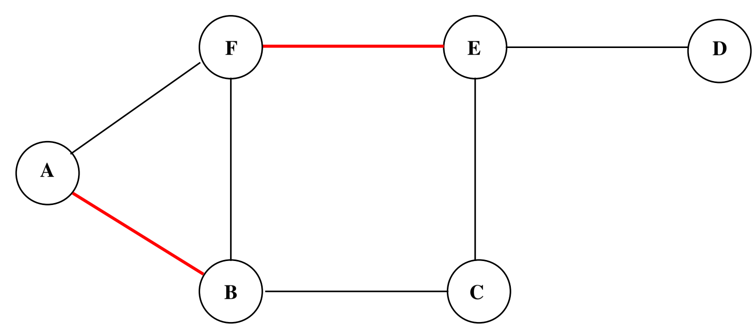


Figure: A matching with 2 edges

Maximum Cardinality Matching Problem

An example:

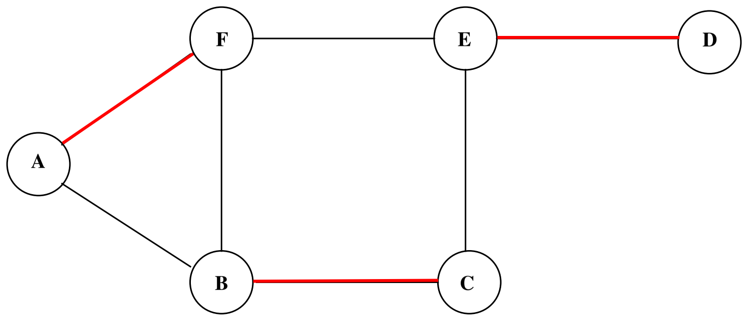


Figure: The maximum matching with 3 edges

Maximum Cardinality Matching Problem

Question: How to write the Maximum Cardinality Matching Problem as a linear programme?

Maximum Cardinality Matching Problem

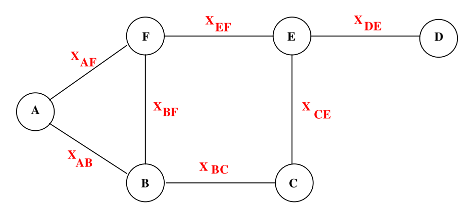
First step: define the variables. Not always easy, most of the time good idea to think of the objective function.

Goal here: find a maximum subset of edges $\rightarrow$ variables on the edges seem useful.

Maximum Cardinality Matching Problem

First step: define the variables. Not always easy, most of the time good idea to think of the objective function.

Goal here: find a maximum subset of edges $\rightarrow$ variables on the edges seem useful.
Variables: one variable per edge, x_{AB} for edge AB.



x_{AB} binary variable $x_{AB} = 1$ if AB in the matching $x_{AB} = 0$ otherwise

Maximum Cardinality Matching Problem

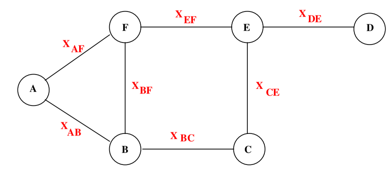
Second step: write the objective function.

$$\max \quad x_{AB} + x_{BC} + \ldots + x_{AF}$$

Maximum Cardinality Matching Problem

Third step: write the contraints.

- A matching $M \subseteq E$ is a collection of edges such that every vertex of V is incident to at most one edge of M.

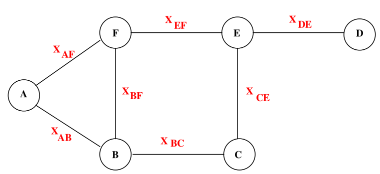


Constraint on vertex A: $x_{AB} + x_{AF} \leq 1$.
Constraint on vertex B: $x_{AB} + x_{BC} + x_{BF} \leq 1$.
One constraint per vertex.

Maximum Cardinality Matching Problem

Third step: write the contraints.

- A matching $M \subseteq E$ is a collection of edges such that every vertex of V is incident to at most one edge of M.

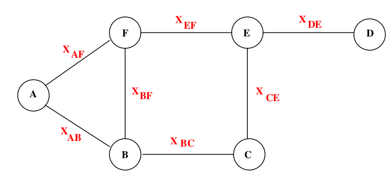


Constraint on vertex A: $x_{AB} + x_{AF} \leq 1$.
Constraint on vertex B: $x_{AB} + x_{BC} + x_{BF} \leq 1$.
One constraint per vertex.

Maximum Cardinality Matching Problem

Third step: write the contraints.

- A matching $M \subseteq E$ is a collection of edges such that every vertex of V is incident to at most one edge of M.

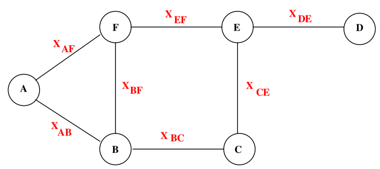


Constraint on vertex A: $x_{AB} + x_{AF} \leq 1$.
Constraint on vertex B: $x_{AB} + x_{BC} + x_{BF} \leq 1$.
One constraint per vertex.

Maximum Cardinality Matching Problem

Third step: write the contraints.

- A matching $M \subseteq E$ is a collection of edges such that every vertex of V is incident to at most one edge of M.



Constraint on vertex A: $x_{AB} + x_{AF} \leq 1$.
Constraint on vertex B: $x_{AB} + x_{BC} + x_{BF} \leq 1$.
One constraint per vertex.

Maximum Cardinality Matching Problem

Variables: $x_{AB} = 1$ if edge AB is in the matching and $x_{AB} = 0$ otherwise.

$$
\begin{aligned}
\max \quad & x_{AB} + x_{BC} + x_{CE} + x_{DE} + x_{EF} + x_{AF} + x_{BF} \\
\text{subject to} \quad & \\
& x_{AB} + x_{AF} \leq 1 \\
& x_{AB} + x_{BC} + x_{BF} \leq 1 \\
& x_{BC} + x_{CE} \leq 1 \\
& x_{DE} \leq 1 \\
& x_{CE} + x_{EF} + x_{DE} \leq 1 \\
& x_{BF} + x_{EF} + x_{AF} \leq 1 \\
& x_{AB}, x_{BC}, x_{CE}, x_{DE}, x_{EF}, x_{AF}, x_{BF} \geq 0 \\
& x_{AB}, x_{BC}, x_{CE}, x_{DE}, x_{EF}, x_{AF}, x_{BF} \in \mathbb{N}
\end{aligned}
$$

Maximum Cardinality Matching Problem

Can be written in a more concise form and more generally for any graph.

$$
\begin{array}{ll}
\text{Var.:} & x_{AB} = 1 \text{ if } AB \in M, \\
 & x_{AB} = 0 \text{ otherwise} \\
\max & x_{AB} + x_{BC} + x_{CE} \\
 & + x_{DE} + x_{EF} + x_{AF} + x_{BF} \\
\text{s.t.} & \\
 & x_{AB} + x_{AF} \leq 1 \\
 & x_{AB} + x_{BC} + x_{BF} \leq 1 \\
 & x_{BC} + x_{CE} \leq 1 \\
 & x_{DE} \leq 1 \\
 & x_{CE} + x_{EF} + x_{DE} \leq 1 \\
 & x_{BF} + x_{EF} + x_{AF} \leq 1 \\
 & x_{AB}, x_{BC}, x_{CE}, x_{DE}, x_{EF}, x_{AF}, x_{BF} \geq 0 \\
 & x_{AB}, x_{BC}, x_{CE}, x_{DE}, x_{EF}, x_{AF}, x_{BF} \in \mathbb{N}
\end{array}
$$

$$
\begin{array}{lll}
\text{Var.:} & x_{ij} = 1 \text{ if } ij \in M, & \\
 & x_{ij} = 0 \text{ otherwise} & \\
\max & \sum_{(i,j) \in E} x_{ij} & \\
\text{s. t.} & & \\
 & \sum_{ij \in E} x_{ij} \leq 1 & (\forall i \in V) \\
 & x_{ij} \geq 0 & (\forall (i,j) \in E) \\
 & x_{ij} \in \mathbb{N} & (\forall (i,j) \in E)
\end{array}
$$

The Simplex Method or Solving Linear Program

Frédéric Giroire

Motivation

- Most popular method to solve linear programs.
- Principle: smartly explore basic solutions (corner point solutions), improving the value of the solution.

The simplex

Start with a problem written under the standard form.

$$\begin{array}{lccccccc}
\text{Maximize} & 5x_1 & + & 4x_2 & + & 3x_3 & & \\
\text{Subject to :} & & & & & & & \\
 & 2x_1 & + & 3x_2 & + & x_3 & \leq & 5 \\
 & 4x_1 & + & x_2 & + & 2x_3 & \leq & 11 \\
 & 3x_1 & + & 4x_2 & + & 2x_3 & \leq & 8 \\
 & & & & & x_1, x_2, x_3 & \geq & 0.
\end{array}$$

The simplex

First step: introduce new variables, slack variables.

$$2x_1 + 3x_2 + x_3 \leq 5$$

We note x_4 the slack (difference) between the right member and 5, that is

$$x_4 = 5 - 2x_1 - 3x_2 - x_3.$$

The inequation can now be written as

$$x_4 \geq 0.$$

The simplex

Similarly, for the 2 others inequalities:

$$\begin{array}{rcrcrcl} 4x_1 & + & x_2 & + & 2x_3 & \leq & 11 \\ 3x_1 & + & 4x_2 & + & 2x_3 & \leq & 8 \end{array}$$

We define x_5 and x_6:

$$\begin{array}{rcrcrcrcr} x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \end{array}$$

And the inequalities can be written as

$$x_5 \geq 0, x_6 \geq 0.$$

The simplex

To summarize, we introduce three slack variables x_4, x_5, x_6:

$$\begin{array}{rcrcrcrcr} x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\ x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\ z & = & & & 5x_1 & + & 4x_2 & + & 3x_3. \end{array}$$

The problem can be written as:

Maximize z subject to $x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$.

slack variables x_4, x_5, x_6 decision variables x_1, x_2, x_3. The two problems are equivalent.

The simplex

Second step: Find an initial solution.

In our example, $x_1 = 0, x_2 = 0, x_3 = 0$ is feasible.

We compute the value of x_4, x_5, x_6.

$$x_4 = 5 - 2x_1 - 3x_2 - x_3 = 5$$

Similarly, $x_5 = 11$ and $x_6 = 8$.

We get an initial solution

$$x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 5, x_5 = 11, x_6 = 8$$

of value $z = 0$

The simplex

Dictionary:

$$\begin{array}{rcrcrcrcr} x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\ x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\ \hline z & = & & & 5x_1 & + & 4x_2 & + & 3x_3. \end{array}$$

Basic variables: x_4, x_5, x_6, variables on the left.
Non-basic variable: x_1, x_2, x_3, variables on the right.

A dictionary is feasible if a feasible solution is obtained by setting all non-basic variables to 0.

The simplex

Dictionary:

$$\begin{array}{rcrcrcrcr} x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\ x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\ \hline z & = & & & 5x_1 & + & 4x_2 & + & 3x_3. \end{array}$$

Basic variables: x_4, x_5, x_6, variables on the left.
Non-basic variable: x_1, x_2, x_3, variables on the right.

A dictionary is feasible if a feasible solution is obtained by setting all non-basic variables to 0.

The simplex

Simplex strategy: find an optimal solution by successive improvements.
Rule: we increase the value of the variable of largest positive coefficient in z.

$$\begin{array}{rcrcrcrcr} x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\ x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\ \hline z & = & & & \boxed{5}x_1 & + & 4x_2 & + & 3x_3. \end{array}$$

Here, we try to increase x_1.

The simplex

How much can we increase x_1?

$$\begin{array}{rcrcrcrcr} x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\ x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\ \hline z & = & & & 5x_1 & + & 4x_2 & + & 3x_3. \end{array}$$

We have $\boxed{x_4 \geq 0}$.

It implies $5 - 2x_1 \geq 0$, that is $\boxed{x_1 \leq \frac{5}{2}}$.

The simplex

How much can we increase x_1?

$$\begin{array}{rcrcrcrcr} x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\ x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\ \hline z & = & & & 5x_1 & + & 4x_2 & + & 3x_3. \end{array}$$

We have $\boxed{x_4 \geq 0}$.
It implies $5 - 2x_1 \geq 0$, that is $\boxed{x_1 \leq 5/2}$.

Similarly,
$x_5 \geq 0$ gives $x_1 \leq 11/4$.
$x_6 \geq 0$ gives $x_1 \leq 8/3$.

The simplex

How much can we increase x_1?

$$\begin{array}{rclcrcrcr} x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\ x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\ \hline z & = & & & 5x_1 & + & 4x_2 & + & 3x_3. \end{array}$$

We have $\boxed{x_4 \geq 0}$.

It implies $5 - 2x_1 \geq 0$, that is $\boxed{x_1 \leq 5/2}$ Strongest constraint

Similarly,
$x_5 \geq 0$ gives $x_1 \leq 11/4$.
$x_6 \geq 0$ gives $x_1 \leq 8/3$.

The simplex

How much can we increase x_1?

$$\begin{array}{rclcrcrcr} x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\ x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\ \hline z & = & & & 5x_1 & + & 4x_2 & + & 3x_3. \end{array}$$

We have $\boxed{x_4 \geq 0}$.

It implies $5 - 2x_1 \geq 0$, that is $\boxed{x_1 \leq 5/2}$ Strongest constraint

We get a new solution: $x_1 = 5/2$, $x_4 = 0$
with better value $z = 5 \cdot 5/2 = 25/2$.
We still have $x_2 = x_3 = 0$ and now $x_5 = 11 - 4 \cdot 5/2 = 1$,
$x_6 = 8 - 3 \cdot 5/2 = 1/2$

The simplex

We build a new feasible dictionary.

$$\begin{array}{rclcrcrcr} x_4 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 \\ x_5 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 \\ \hline z & = & & & 5x_1 & + & 4x_2 & + & 3x_3. \end{array}$$

x_1 enters the bases and x_4 leaves it:

$$x_1 = 5/2 - 3/2x_2 - 1/2x_3 - 1/2x_4$$

The simplex

We replace x_1 by its expression in function of x_2, x_3, x_4.

$$\begin{array}{rclcrcrcr} x_1 & = & 5/2 & - & 1/2x_4 & - & 3/2x_2 & - & 1/2x_3 \\ x_5 & = & 11 & - & 4(5/2-3/2x_2-1/2x_3-1/2x_4) & - & x_2 & - & 2x_3 \\ x_6 & = & 8 & - & 3(5/2-3/2x_2-1/2x_3-1/2x_4) & - & 4x_2 & - & 2x_3 \\ \hline z & = & & & 5(5/2-3/2x_2-1/2x_3-1/2x_4) & + & 4x_2 & + & 3x_3. \end{array}$$

The simplex

Finally, we get the new dictionary:

$$\begin{array}{rcl} x_1 & = & \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 & = & 1 + 5x_2 + 2x_4 \\ x_6 & = & \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ \hline z & = & \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4. \end{array}$$

The simplex

Finally, we get the new dictionary:

$$\begin{array}{rcl} x_1 & = & \boxed{5/2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 & = & \boxed{1} + 5x_2 + 2x_4 \\ x_6 & = & \boxed{1/2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ \hline z & = & \boxed{25/2} - 7/2\,x_2 + 1/2\,x_3 - 5/2\,x_4. \end{array}$$

We can read the solution directly from the dictionary:
Non basic variables: $x_2 = x_3 = x_4 = 0$.
Basic variables: $x_1 = 5/2$, $x_5 = 1$, $x_6 = 1/2$.
Value of the solution: $z = 25/2$.

The simplex

$$\begin{array}{rcl} x_1 & = & \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 & = & 1 + 5x_2 + 2x_4 \\ x_6 & = & \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ \hline z & = & \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4. \end{array}$$

New step of the simplex:

- x_3 enters the basis (variable with largest positive coefficient).
- 3^d equation is the strictest constaint $x_3 \leq 1$.
- x_6 leaves the basis.

The simplex

New feasible dictionary:

$$\begin{array}{rcl} x_3 & = & 1 + x_2 + 3x_4 - 2x_6 \\ x_1 & = & 2 - 2x_2 - 2x_4 + x_6 \\ x_5 & = & 1 + 5x_2 + 2x_4 \\ \hline z & = & 13 - 3x_2 - x_4 - x_6. \end{array}$$

With new solution:

$$x_1 = 2, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 0$$

of value $z = 13$.

This solution is optimal.
All coefficients in z are negative and $x_2 \geq 0, x_4 \geq 0, x_6 \geq 0$, so $z \leq 13$.

The simplex

New feasible dictionary:

$$\begin{array}{rcrcrcrcr} x_3 & = & 1 & + & x_2 & + & 3x_4 & - & 2x_6 \\ x_1 & = & 2 & - & 2x_2 & - & 2x_4 & + & x_6 \\ x_5 & = & 1 & + & 5x_2 & + & 2x_4 & & \\ \hline z & = & 13 & - & 3x_2 & - & x_4 & - & x_6. \end{array}$$

With new solution:

$$x_1 = 2, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 0$$

of value $z = 13$.

$\boxed{\text{This solution is optimal.}}$
All coefficients in z are negative and $x_2 \geq 0, x_4 \geq 0, x_6 \geq 0$*, so* $z \leq 13$.

Take Aways

- Most popular method to solve linear programs.
- Principle: smartly explore basic solutions (corner point solutions), improving the value of the solution.
- Complexity:
  - In theory, NP-complete (can explore a number of solutions exponentiel in the number of variables and constraints).
  - In practice, almost linear in the number of constraints.
- Polynomial methods exists: the ellipsoid method.

Linear Program Duality

Frédéric Giroire

Motivation

- Finding bounds on the optimal solution. Provides a measure of the "goodness" of a solution.
- Provide certificate of optimality.
- Economic interpretation of the dual problem.

**** Introduction to Duality ****

Duality Theorem: introduction

$$
\begin{array}{lrcrcrcrcl}
\text{Maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\
\text{Subject to :} & & & & & & & & & \\
 & x_1 & - & x_2 & - & x_3 & + & 3x_4 & \leq & 1 \\
 & 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \leq & 55 \\
 & -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \leq & 3 \\
 & & & & & & & x_1, x_2, x_3, x_4 & \geq & 0.
\end{array}
$$

Lower bound: a feasible solution, e.g. $(0, 0, 1, 0) \Rightarrow z^* \geq 5$.

What if we want an upper bound?

Duality Theorem: introduction

$$\begin{array}{lrcrcrcrcl}
\text{Maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\
\text{Subject to :} & & & & & & & & & \\
& x_1 & - & x_2 & - & x_3 & + & 3x_4 & \leq & 1 \\
& 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \leq & 55 \\
& -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \leq & 3 \\
& & & & & & & x_1, x_2, x_3, x_4 & \geq & 0.
\end{array}$$

Second Inequation $\times 5/3$:

$$\frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4 \leq \frac{275}{3}.$$

Note that (all variables are positive),

$$4x_1 + x_2 + 5x_3 + 3x_4 \leq \frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4$$

Hence, a first bound:

$$z^* \leq \frac{275}{3}.$$

Duality Theorem: introduction

$$\begin{array}{lrcrcrcrcl}
\text{Maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\
\text{Subject to :} & & & & & & & & & \\
& x_1 & - & x_2 & - & x_3 & + & 3x_4 & \leq & 1 \\
& 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \leq & 55 \\
& -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \leq & 3 \\
& & & & & & & x_1, x_2, x_3, x_4 & \geq & 0.
\end{array}$$

Second Inequation $\times 5/3$:

$$\frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4 \leq \frac{275}{3}.$$

Note that (all variables are positive),

$$4x_1 + x_2 + 5x_3 + 3x_4 \leq \frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4$$

Hence, a first bound:

$$z^* \leq \frac{275}{3}.$$

Duality Theorem: introduction

$$\begin{array}{lrcrcrcrcl}
\text{Maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\
\text{Subject to :} & & & & & & & & & \\
& x_1 & - & x_2 & - & x_3 & + & 3x_4 & \leq & 1 \\
& 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \leq & 55 \\
& -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \leq & 3 \\
& & & & & & & x_1, x_2, x_3, x_4 & \geq & 0.
\end{array}$$

Second Inequation $\times 5/3$:

$$\frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4 \leq \frac{275}{3}.$$

Note that (all variables are positive),

$$4x_1 + x_2 + 5x_3 + 3x_4 \leq \frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4$$

Hence, a first bound:

$$z^* \leq \frac{275}{3}.$$

Duality Theorem: introduction

$$\begin{array}{lrcrcrcrcl}
\text{Maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\
\text{Subject to :} & & & & & & & & & \\
& x_1 & - & x_2 & - & x_3 & + & 3x_4 & \leq & 1 \\
& 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \leq & 55 \\
& -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \leq & 3 \\
& & & & & & & x_1, x_2, x_3, x_4 & \geq & 0.
\end{array}$$

Similarly, $2^d + 3^d$ constraints:

$$4x_1 + 3x_2 + 6x_3 + 3x_4 \leq 58.$$

Hence, a second bound:

$$z^* \leq 58.$$

$\rightarrow$ need for a systematic strategy.

Duality Theorem: introduction

$$
\begin{array}{lccccccccccc}
\text{Maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\
\text{Subject to :} & & & & & & & & & \\
& x_1 & - & x_2 & - & x_3 & + & 3x_4 & \le & 1 \\
& 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \le & 55 \\
& -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \le & 3 \\
& & & & & & & x_1, x_2, x_3, x_4 & \ge & 0.
\end{array}
$$

Similarly, $2^d + 3^d$ constraints:

$$4x_1 + 3x_2 + 6x_3 + 3x_4 \le 58.$$

Hence, a second bound:

$$z^* \le 58.$$

$\rightarrow$ need for a systematic strategy.

Duality Theorem: introduction

$$
\begin{array}{lccccccccccc}
\text{Maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & & \\
\text{Subject to :} & & & & & & & & & & \\
& x_1 & - & x_2 & - & x_3 & + & 3x_4 & \le & 1 & \boxed{\times y_1} \\
& 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \le & 55 & \boxed{\times y_2} \\
& -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \le & 3 & \boxed{\times y_3} \\
& & & & & & & x_1, x_2, x_3, x_4 & \ge & 0. &
\end{array}
$$

Build linear combinations of the constraints. Summing:

$$(y_1 + 5y_2 - y_3)x_1 + (-y_1 + y_2 + 2y_3)x_2 + (-y_1 + 3y_2 + 3y_3)x_3$$
$$+ (3y_1 + 8y_2 - 5y_3)x_4 \le y_1 + 55y_2 + 3y_3.$$

We want left part upper bound of z. We need coefficient of $x_j \ge$ coefficient in z:

$$
\begin{array}{ccccccc}
y_1 & + & 5y_2 & - & y_3 & \ge & 4 \\
-y_1 & + & y_2 & + & 2y_3 & \ge & 1 \\
-y_1 & + & 3y_2 & + & 3y_3 & \ge & 5 \\
3y_1 & + & 8y_2 & - & 5y_3 & \ge & 3.
\end{array}
$$

Duality Theorem: introduction

Build linear combinations of the constraints. Summing:

$$(y_1 + 5y_2 - y_3)x_1 + (-y_1 + y_2 + 2y_3)x_2 + (-y_1 + 3y_2 + 3y_3)x_3$$
$$+ (3y_1 + 8y_2 - 5y_3)x_4 \le y_1 + 55y_2 + 3y_3.$$

We want left part upper bound of z. We need coefficient of $x_j \ge$ coefficient in z:

$$
\begin{array}{ccccccc}
y_1 & + & 5y_2 & - & y_3 & \ge & 4 \\
-y_1 & + & y_2 & + & 2y_3 & \ge & 1 \\
-y_1 & + & 3y_2 & + & 3y_3 & \ge & 5 \\
3y_1 & + & 8y_2 & - & 5y_3 & \ge & 3.
\end{array}
$$

If the $y_i \ge 0$ and satisfy theses inequations, then

$$4x_1 + x_2 + 5x_3 + 3x_4 \le y_1 + 55y_2 + 3y_3.$$

In particular,

$$z^* \le y_1 + 55y_2 + 3y_3.$$

Duality Theorem: introduction

Objective: smallest possible upper bound. Hence, we solve the following PL:

$$
\begin{array}{lccccccc}
\text{Minimize} & y_1 & + & 55y_2 & + & 3y_3 & & \\
\text{Subject to:} & & & & & & & \\
& y_1 & + & 5y_2 & - & y_3 & \ge & 4 \\
& -y_1 & + & y_2 & + & 2y_3 & \ge & 1 \\
& -y_1 & + & 3y_2 & + & 3y_3 & \ge & 5 \\
& 3y_1 & + & 8y_2 & - & 5y_3 & \ge & 3 \\
& & & & & y_1, y_2, y_3 & \ge & 0.
\end{array}
\qquad (1)
$$

It is the dual problem of the problem.

Duality Theorem: introduction

Primal Linear Program

$$\begin{array}{llllllllll}
\text{Maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\
\text{Subject to :} & x_1 & - & x_2 & - & x_3 & + & 3x_4 & \leq & 1 \\
 & 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \leq & 55 \\
 & -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \leq & 3 \\
 & & & & & & & x_1, x_2, x_3, x_4 & \geq & 0.
\end{array}$$

Dual Linear Program

$$\begin{array}{llllllll}
\text{Minimize} & y_1 & + & 55y_2 & + & 3y_3 & & \\
\text{Subject to:} & y_1 & + & 5y_2 & - & y_3 & \geq & 4 \\
 & -y_1 & + & y_2 & + & 2y_3 & \geq & 1 \\
 & -y_1 & + & 3y_2 & + & 3y_3 & \geq & 5 \\
 & 3y_1 & + & 8y_2 & - & 5y_3 & \geq & 3 \\
 & & & & & y_1, y_2, y_3 & \geq & 0.
\end{array} \quad (2)$$

Exercise. Writing the dual.

Duality Theorem: introduction

Primal	Dual
3 constraints	3 variables
4 variables (columns)	4 constraints (raws)
$\leq$	$\geq$
Maximize	Minimize
Coef objective function	Right member of constraints
Right member of constraints	Coef objective function
A	A^t

**** Duality ****

The Dual Problem

Primal problem:

$$\begin{array}{llll}
\text{Maximize} & \sum_{j=1}^{n} c_j x_j & & \\
\text{Subject to:} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i & (i = 1, 2, \cdots, m) & \\
 & x_j \geq 0 & (j = 1, 2, \cdots, n) &
\end{array} \quad (3)$$

Its dual problem is defined by the LP problem:

$$\begin{array}{llll}
\text{Minimize} & \sum_{i=1}^{m} b_i y_i & & \\
\text{Subject to:} & \sum_{i=1}^{m} a_{ij} y_i \geq c_j & (j = 1, 2, \cdots, n) & \\
 & y_i \geq 0 & (i = 1, 2, \cdots, m) &
\end{array} \quad (4)$$

Weak Duality Theorem

Theorem: If $(x_1, x_2, \ldots, x_n)$ is feasible for the primal and $(y_1, y_2, \ldots y_m)$ is feasible for the dual, then

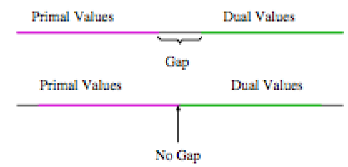
$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i.$$

Proof:

$$\begin{aligned} \textstyle\sum_j c_j x_j &\leq \textstyle\sum_j (\sum_i y_i a_{ij}) x_j && \text{dual definition: } \textstyle\sum_i y_i a_{ij} \geq c_j \\ &= \textstyle\sum_i (\sum_j a_{ij} x_j) y_i && \\ &\leq \textstyle\sum_i b_i y_i && \text{primal definition: } \textstyle\sum_i x_i a_{ij} \leq b_j \end{aligned}$$

Gap or No Gap?

An important question:
Is there a gap between the largest primal value and the smallest dual value?



Strong Duality Theorem

Theorem: If the primal problem has an optimal solution,

$$x^* = (x_1^*, \ldots, X_n^*),$$

then the dual also has an optimal solution,

$$y^* = (y_1^*, \ldots, y_n^*),$$

and they have same optimal value, i.e.

$$\sum_j c_j x_j^* = \sum_i b_i y_i^*.$$

Relationship between the Primal and Dual Problems

Lemma: The dual of the dual is always the primal problem.
Corollary: + (Strong Duality Theorem) $\Rightarrow$ Primal has an optimal solution iff dual has an optimal solution.
Weak duality: Primal unbounded $\Rightarrow$ dual unfeasible.

		Dual		
		Optimal	Unfeasible	Unbounded
	Optimal	X		
Primal	Unfeasible		X	X
	Unbounded		X	

Relationship between the Primal and Dual Problems

Lemma: The dual of the dual is always the primal problem.
Corollary: + (Strong Duality Theorem) $\Rightarrow$ Primal has an optimal solution iff dual has an optimal solution.
Weak duality: Primal unbounded $\Rightarrow$ dual unfeasible.

		Dual		
		Optimal	Unfeasible	Unbounded
	Optimal	X		
Primal	Unfeasible		X	X
	Unbounded		X	

**** Certificate of Optimality ****

Complementary Slackness

Theorem: Let $x_1^*, \dots x_n^*$ be a feasible solution of the primal and $y_1^*, \dots y_n^*$ be a feasible solution of the dual. Then,

$$\sum_{i=1}^{m} a_{ij} y_i^* = c_j \quad \text{or} \quad x_j^* = 0 \quad \text{or both} (j = 1, 2, \dots n)$$

$$\sum_{j=1}^{n} a_{ij} x_j^* = b_i \quad \text{or} \quad y_i^* = 0 \quad \text{or both} (i = 1, 2, \dots m)$$

are necessary and sufficient conditions to have the optimality of x^* and y^*.

Complementary Slackness - Proof

x^* feasible $\Rightarrow b_i - \sum_j a_{ij} x_j \geq 0$.
y^* dual feasible, hence non negative.

Thus

$$(b_i - \sum_j a_{ij} x_j) y_i \geq 0.$$

Similarly,
y^* dual feasible $\Rightarrow \sum_i a_{ij} y_i - c_j \geq 0$.
x^* feasible, hence non negative.

$$(\sum_i a_{ij} y_i - c_j) x_j \geq 0.$$

Complementary Slackness - Proof

$$(b_i - \sum_j a_{ij}x_j)y_i \geq 0 \qquad \text{and} \qquad (\sum_i a_{ij}y_i - c_j)x_j \geq 0$$

By summing, we get:

$$\sum_i (b_i - \sum_j a_{ij}x_j)y_i \geq 0 \qquad \text{and} \qquad \sum_j (\sum_i a_{ij}y_i - c_j)x_j \geq 0$$

Summing + strong duality theorem:

$$\sum_i b_i y_i - \sum_{i,j} a_{ij}x_j y_i + \sum_{j,i} a_{ij}y_i x_j - \sum_j c_j x_j = \sum_i b_i y_i - \sum c_j x_j = 0.$$

Implies: inequalities must be equalities:

$$\forall i, (b_i - \sum_j a_{ij}x_j)y_i = 0 \qquad \text{and} \qquad \forall j (\sum_i a_{ij}y_i - c_j)x_j = 0.$$

$XY = 0$ if $X = 0$ or $Y = 0$. Done.

Complementary Slackness - Proof

$$(b_i - \sum_j a_{ij}x_j)y_i \geq 0 \qquad \text{and} \qquad (\sum_i a_{ij}y_i - c_j)x_j \geq 0$$

By summing, we get:

$$\sum_i (b_i - \sum_j a_{ij}x_j)y_i \geq 0 \qquad \text{and} \qquad \sum_j (\sum_i a_{ij}y_i - c_j)x_j \geq 0$$

Summing + strong duality theorem:

$$\sum_i b_i y_i - \sum_{i,j} a_{ij}x_j y_i + \sum_{j,i} a_{ij}y_i x_j - \sum_j c_j x_j = \sum_i b_i y_i - \sum c_j x_j = 0.$$

Implies: inequalities must be equalities:

$$\forall i, (b_i - \sum_j a_{ij}x_j)y_i = 0 \qquad \text{and} \qquad \forall j (\sum_i a_{ij}y_i - c_j)x_j = 0.$$

$XY = 0$ if $X = 0$ or $Y = 0$. Done.

FG COATI Duality 20/27

Complementary Slackness - Proof

$$(b_i - \sum_j a_{ij}x_j)y_i \geq 0 \qquad \text{and} \qquad (\sum_i a_{ij}y_i - c_j)x_j \geq 0$$

By summing, we get:

$$\sum_i (b_i - \sum_j a_{ij}x_j)y_i \geq 0 \qquad \text{and} \qquad \sum_j (\sum_i a_{ij}y_i - c_j)x_j \geq 0$$

Summing + strong duality theorem:

$$\sum_i b_i y_i - \sum_{i,j} a_{ij}x_j y_i + \sum_{j,i} a_{ij}y_i x_j - \sum_j c_j x_j = \sum_i b_i y_i - \sum c_j x_j = 0.$$

Implies: inequalities must be equalities:

$$\forall i, (b_i - \sum_j a_{ij}x_j)y_i = 0 \qquad \text{and} \qquad \forall j (\sum_i a_{ij}y_i - c_j)x_j = 0.$$

$XY = 0$ if $X = 0$ or $Y = 0$. Done.

FG COATI Duality 20/27

Complementary Slackness - Proof

$$(b_i - \sum_j a_{ij}x_j)y_i \geq 0 \qquad \text{and} \qquad (\sum_i a_{ij}y_i - c_j)x_j \geq 0$$

By summing, we get:

$$\sum_i (b_i - \sum_j a_{ij}x_j)y_i \geq 0 \qquad \text{and} \qquad \sum_j (\sum_i a_{ij}y_i - c_j)x_j \geq 0$$

Summing + strong duality theorem:

$$\sum_i b_i y_i - \sum_{i,j} a_{ij}x_j y_i + \sum_{j,i} a_{ij}y_i x_j - \sum_j c_j x_j = \sum_i b_i y_i - \sum c_j x_j = 0.$$

Implies: inequalities must be equalities:

$$\forall i, (b_i - \sum_j a_{ij}x_j)y_i = 0 \qquad \text{and} \qquad \forall j (\sum_i a_{ij}y_i - c_j)x_j = 0.$$

$XY = 0$ if $X = 0$ or $Y = 0$. Done.

FG COATI Duality 20/27

Theorem [Optimality Certificate]: A feasible solution $x_1^*, \ldots x_n^*$ of the primal is **optimal** iif there exist numbers $y_1^*, \ldots y_n^*$ such that

1. they **satisfy the complementary slackness** condition:

$$\begin{array}{lll} \sum_{i=1}^m a_{ij} y_i^* & = c_j & \text{when } x_j^* > 0 \\ y_j^* & = 0 & \text{when } \sum_{j=1}^n a_{ij} x_j^* < b_i \end{array}$$

2. and $y_1^*, \ldots y_n^*$ **feasible solution of the dual**, that is

$$\begin{array}{lll} \sum_{i=1}^m a_{ij} y_i^* & \geq c_j & \forall j = 1, \ldots n \\ y_i^* & \geq 0 & \forall i = 1, \ldots, m. \end{array}$$

Example: Verify that $(2,4,0,0,7,0)$ optimal solution of

$$\begin{array}{lrcrcrcrcrcrcl} \text{Max} & 18x_1 & - & 7x_2 & + & 12x_3 & + & 5x_4 & & & + & 8x_6 & & \\ \text{st:} & 2x_1 & - & 6x_2 & + & 2x_3 & + & 7x_4 & + & 3x_5 & + & 8x_6 & \leq & 1 \\ & -3x_1 & - & x_2 & + & 4x_3 & - & 3x_4 & + & x_5 & + & 2x_6 & \leq & -2 \\ & 8x_1 & - & 3x_2 & + & 5x_3 & - & 2x_4 & & & + & 2x_6 & \leq & 4 \\ & 4x_1 & & & + & 8x_3 & + & 7x_4 & - & x_5 & + & 3x_6 & \leq & 1 \\ & 5x_1 & + & 2x_2 & - & 3x_3 & + & 6x_4 & - & 2x_5 & - & x_6 & \leq & 5 \\ & & & & & & & & & & & x_1, x_2, \cdots, x_6 & \geq & 0 \end{array}$$

First step: Existence of $y_1^*, \ldots, y_5^*$, such as

$$\begin{array}{lll} \sum_{i=1}^m a_{ij} y_i^* & = c_j & \text{when } x_j^* > 0 \\ y_i^* & = 0 & \text{when } \sum_{j=1}^n a_{ij} x_j^* < b_i \end{array}$$

That is

$$\begin{array}{rcrcrcrcrcl} 2y_1^* & - & 3y_2^* & + & 8y_3^* & + & 4y_4^* & + & 5y_5^* & = & 18 \\ -6y_1^* & - & y_2^* & - & 3y_3^* & & & + & 2y_5^* & = & -7 \\ 3y_1^* & + & y_2^* & & & - & y_4^* & - & 2y_5^* & = & 0 \\ & & y_2^* & & & & & & & = & 0 \\ & & & & & & & & y_5^* & = & 0 \end{array}$$

$(\frac{1}{3}, 0, \frac{5}{3}, 1, 0)$ is solution.

Example: Verify that $(2,4,0,0,7,0)$ optimal solution of

$$\begin{array}{lrcrcrcrcrcrcl} \text{Max} & 18x_1 & - & 7x_2 & + & 12x_3 & + & 5x_4 & & & + & 8x_6 & & \\ \text{st:} & 2x_1 & - & 6x_2 & + & 2x_3 & + & 7x_4 & + & 3x_5 & + & 8x_6 & \leq & 1 \\ & -3x_1 & - & x_2 & + & 4x_3 & - & 3x_4 & + & x_5 & + & 2x_6 & \leq & -2 \\ & 8x_1 & - & 3x_2 & + & 5x_3 & - & 2x_4 & & & + & 2x_6 & \leq & 4 \\ & 4x_1 & & & + & 8x_3 & + & 7x_4 & - & x_5 & + & 3x_6 & \leq & 1 \\ & 5x_1 & + & 2x_2 & - & 3x_3 & + & 6x_4 & - & 2x_5 & - & x_6 & \leq & 5 \\ & & & & & & & & & & & x_1, x_2, \cdots, x_6 & \geq & 0 \end{array}$$

Second step: Verify $(\frac{1}{3}, 0, \frac{5}{3}, 1, 0)$ is a solution of the dual.

$$\begin{array}{lll} \sum_{i=1}^m a_{ij} y_i^* & \geq c_j & \forall j = 1, \ldots n \\ y_j^* & \geq 0 & \forall i = 1, \ldots, m. \end{array}$$

Example: Verify that $(2,4,0,0,7,0)$ optimal solution of

$$\begin{array}{lrcrcrcrcrcrcl} \text{Max} & 18x_1 & - & 7x_2 & + & 12x_3 & + & 5x_4 & & & + & 8x_6 & & \\ \text{st:} & 2x_1 & - & 6x_2 & + & 2x_3 & + & 7x_4 & + & 3x_5 & + & 8x_6 & \leq & 1 \\ & -3x_1 & - & x_2 & + & 4x_3 & - & 3x_4 & + & x_5 & + & 2x_6 & \leq & -2 \\ & 8x_1 & - & 3x_2 & + & 5x_3 & - & 2x_4 & & & + & 2x_6 & \leq & 4 \\ & 4x_1 & & & + & 8x_3 & + & 7x_4 & - & x_5 & + & 3x_6 & \leq & 1 \\ & 5x_1 & + & 2x_2 & - & 3x_3 & + & 6x_4 & - & 2x_5 & - & x_6 & \leq & 5 \\ & & & & & & & & & & & x_1, x_2, \cdots, x_6 & \geq & 0 \end{array}$$

Second step: Verify $(\frac{1}{3}, 0, \frac{5}{3}, 1, 0)$ is a solution of the dual.

$$\begin{array}{lll} \sum_{i=1}^m a_{ij} y_i^* & \geq c_j & \forall j = 1, \ldots n \\ y_j^* & \geq 0 & \forall i = 1, \ldots, m. \end{array}$$

That is, we check

$$\begin{array}{rcrcrcrcrcl} 2y_1^* & - & 3y_2^* & + & 8y_3^* & + & 4y_4^* & + & 5y_5^* & \geq & 18 \\ -6y_1^* & - & y_2^* & - & 3y_3^* & & & + & 2y_5^* & \geq & -7 \\ 2y_1^* & + & 4y_2^* & + & 5y_3^* & + & 8y^4 & + & 3y_5^* & \geq & 12 \\ 7y_1^* & - & 3y_2^* & - & 2y_3^* & + & 7y^4 & + & 6y_5^* & \geq & 5 \\ 3y_1^* & + & y_2^* & & & - & y_4^* & - & 2y_5^* & \geq & 0 \\ 8y_1^* & + & 2y_2^* & + & 2y_3^* & + & 3y^4 & 1 & y_5^* & \geq & 8 \end{array}$$

Example: Verify that $(2,4,0,0,7,0)$ optimal solution of

$$
\begin{array}{llllllllllllll}
\text{Max} & 18x_1 & - & 7x_2 & + & 12x_3 & + & 5x_4 & & & + & 8x_6 & & \\
\text{st:} & 2x_1 & - & 6x_2 & + & 2x_3 & + & 7x_4 & + & 3x_5 & + & 8x_6 & \leq & 1 \\
 & -3x_1 & - & x_2 & + & 4x_3 & - & 3x_4 & + & x_5 & + & 2x_6 & \leq & -2 \\
 & 8x_1 & - & 3x_2 & + & 5x_3 & - & 2x_4 & & & + & 2x_6 & \leq & 4 \\
 & 4x_1 & & & + & 8x_3 & + & 7x_4 & - & x_5 & + & 3x_6 & \leq & 1 \\
 & 5x_1 & + & 2x_2 & - & 3x_3 & + & 6x_4 & - & 2x_5 & - & x_6 & \leq & 5 \\
 & & & & & & & & & & & x_1, x_2, \cdots, x_6 & \geq & 0
\end{array}
$$

Second step: Verify $(\frac{1}{3}, 0, \frac{5}{3}, 1, 0)$ is a solution of the dual.

$$
\begin{array}{lll}
\sum_{i=1}^m a_{ij} y_i^* & \geq c_j & \forall j = 1, \ldots n \\
y_j^* & \geq 0 & \forall i = 1, \ldots, m.
\end{array}
$$

That is, we check

$$
\begin{array}{llllllllllllll}
2y_1^* & - & 3y_2^* & + & 8y_3^* & + & 4y_4^* & + & 5y_5^* & \geq & 18 & \text{OK} \\
-6y_1^* & - & y_2^* & - & 3y_3^* & & & + & 2y_5^* & \geq & -7 & \text{OK} \\
2y_1^* & + & 4y_2^* & + & 5y_3^* & + & 8y^4 & + & 3y_5^* & \geq & 12 & \\
7y_1^* & - & 3y_2^* & - & 2y_3^* & + & 7y^4 & + & 6y_5^* & \geq & 5 & \\
3y_1^* & + & y_2^* & & & - & y_4^* & - & 2y_5^* & \geq & 0 & \text{OK} \\
8y_1^* & + & 2y_2^* & + & 2y_3^* & + & 3y^4 & 1 & y_5^* & \geq & 8 &
\end{array}
$$

Only three equations to check.

Example: Verify that $(2,4,0,0,7,0)$ optimal solution of

$$
\begin{array}{llllllllllllll}
\text{Max} & 18x_1 & - & 7x_2 & + & 12x_3 & + & 5x_4 & & & + & 8x_6 & & \\
\text{st:} & 2x_1 & - & 6x_2 & + & 2x_3 & + & 7x_4 & + & 3x_5 & + & 8x_6 & \leq & 1 \\
 & -3x_1 & - & x_2 & + & 4x_3 & - & 3x_4 & + & x_5 & + & 2x_6 & \leq & -2 \\
 & 8x_1 & - & 3x_2 & + & 5x_3 & - & 2x_4 & & & + & 2x_6 & \leq & 4 \\
 & 4x_1 & & & + & 8x_3 & + & 7x_4 & - & x_5 & + & 3x_6 & \leq & 1 \\
 & 5x_1 & + & 2x_2 & - & 3x_3 & + & 6x_4 & - & 2x_5 & - & x_6 & \leq & 5 \\
 & & & & & & & & & & & x_1, x_2, \cdots, x_6 & \geq & 0
\end{array}
$$

Second step: Verify $(\frac{1}{3}, 0, \frac{5}{3}, 1, 0)$ is a solution of the dual.

$$
\begin{array}{lll}
\sum_{i=1}^m a_{ij} y_i^* & \geq c_j & \forall j = 1, \ldots n \\
y_j^* & \geq 0 & \forall i = 1, \ldots, m.
\end{array}
$$

That is, we check

$$
\begin{array}{llllllllllllll}
2y_1^* & - & 3y_2^* & + & 8y_3^* & + & 4y_4^* & + & 5y_5^* & \geq & 18 & \text{OK} \\
-6y_1^* & - & y_2^* & - & 3y_3^* & & & + & 2y_5^* & \geq & -7 & \text{OK} \\
2y_1^* & + & 4y_2^* & + & 5y_3^* & + & 8y^4 & + & 3y_5^* & \geq & 12 & \\
7y_1^* & - & 3y_2^* & - & 2y_3^* & + & 7y^4 & + & 6y_5^* & \geq & 5 & \\
3y_1^* & + & y_2^* & & & - & y_4^* & - & 2y_5^* & \geq & 0 & \text{OK} \\
8y_1^* & + & 2y_2^* & + & 2y_3^* & + & 3y^4 & 1 & y_5^* & \geq & 8 &
\end{array}
$$

Only three equations to check.

OK. The solution $(2,4,0,0,7,0)$ is optimal for the primal problem.

**** Economical Interpretation ****

Signification of Dual Variables

$$
\begin{array}{llllll}
\text{Maximize} & \sum_{j=1}^n c_j x_j & & & \\
\text{Subject to:} & \sum_{j=1}^n a_{ij} x_j & \leq & b_i & (i = 1, 2, \cdots, m) \\
 & x_j & \geq & 0 & (j = 1, 2, \cdots, n)
\end{array}
\qquad
\begin{array}{llllll}
\text{Minimize} & \sum_{i=1}^m b_i y_i & & & \\
\text{Subject to:} & \sum_{i=1}^m a_{ij} y_i & \geq & c_j & (j = 1, 2, \cdots, n) \\
 & y_i & \geq & 0 & (i = 1, 2, \cdots, m)
\end{array}
$$

Signification can be given to variables of the dual problem (dimension analysis):

- x_j: production of a product j (chair, ...)
- b_i: available quantity of resource i (wood, metal, ...)
- a_{ij}: unit of resource i per unit of product j
- c_j: net benefit of the production of a unit of product j

Signification of Dual Variables

$$\begin{array}{llcll} \text{Maximize} & \sum_{j=1}^n c_j x_j & & & \\ \text{Subject to:} & \sum_{j=1}^n a_{ij} x_j & \leq & b_i & (i = 1, 2, \cdots, m) \\ & x_j & \geq & 0 & (j = 1, 2, \cdots, n) \end{array} \qquad \begin{array}{llcll} \text{Minimize} & \sum_{i=1}^m b_i y_i & & & \\ \text{Subject to:} & \sum_{i=1}^m a_{ij} y_i & \geq & c_j & (j = 1, 2, \cdots, n) \\ & y_i & \geq & 0 & (i = 1, 2, \cdots, m) \end{array}$$

Signification can be given to variables of the dual problem (dimension analysis):

- x_j: production of a product j (chair, ...)
- b_i: available quantity of resource i (wood, metal, ...)
- a_{ij}: unit of resource i per unit of product j
- c_j: net benefit of the production of a unit of product j

unit of resource i/unit of product j   euros/unit of product j   euros/unit of resource i

$$a_{1j} y_1 + \dots + a_{nj} y_n > c_j$$

$\rightarrow y_i$ euro by unit of resource i.
Marginal cost of resource i.

Signification of Dual Variables

Theorem: If the LP admits at least one optimal solution, then there exists $\varepsilon > 0$, with the property: If $|t_i| \leq \varepsilon \ \forall i = 1, 2, \cdots, m$, then the LP

$$\begin{array}{llcll} \text{Max} & \sum_{j=1}^n c_j x_j & & & \\ \text{Subject to:} & \sum_{j=1}^n a_{ij} x_j & \leq & b_i + t_i & (i = 1, 2, \cdots, m) \\ & x_j & \geq & 0 & (j = 1, 2, \cdots, n). \end{array} \qquad (5)$$

has an optimal solution and the optimal value of the objective is

$$z^* + \sum_{i=1}^m y_i^* t_i$$

with z^* the optimal solution of the initial LP and $(y_1^*, y_2^*, \cdots, y_m^*)$ the optimal solution of its dual.

Take Aways

- Writing the dual.
- Weak duality theorem: Feasible solution of the dual is an upper bound on the optimal solution of the primal (maximization).
- Strong duality theorem: Primal and dual have the same optimal value (if they have optimal solutions).
- Certificate of optimality.
- Signification of dual variables.

Approximation Algorithms

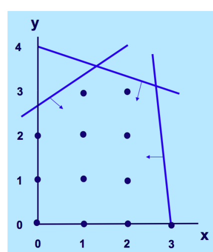
Frédéric Giroire

Motivation

- Goal:
  - Find “good” solutions for difficult problems (NP-hard).
  - Be able to quantify the “goodness” of the given solution.
- Presentation of a technique to get approximation algorithms: fractional relaxation of integer linear programs.

Fractional Relaxation

- Reminder:
  - Integer Linear Programs often hard to solve (NP-hard).
  - Linear Programs (with real numbers) easier to solve (polynomial-time algorithms).
- Idea:
  - 1- Relax the integrality constraints;
  - 2- Solve the (fractional) linear program and then;
  - 3- Round the solution to obtain an integral solution.



Approximation Algorithms

Definition: An approximation algorithm produces

- in polynomial time
- a feasible solution
- whose objective function value is close to the optimal OPT, by close we mean within a guaranteed factor of the optimal.

Example: a factor 2 approximation algorithm for the cardinality vertex cover problem, i.e. an algorithm that finds a cover of cost $\leq 2 \cdot OPT$ in time polynomial in $|V|$.

Set Cover

- Problem: Given a universe U of n elements, a collection of subsets of U, $\mathscr{S} = S_1, \dots, S_k$, and a cost function $c : S \rightarrow Q^+$, find a minimum cost subcollection of S that covers all elements of U.
- Model numerous classical problems as special cases of set cover: vertex cover, minimum cost shortest path...
- Definition: The frequency of an element is the number of sets it is in. The frequency of the most frequent element is denoted by f.
- Various approximation algorithms for set cover achieve one of the two factors $O(\log n)$ or f.

Fractional relaxation

Write a linear program to solve set cover (and its fractional relaxation).

Fractional relaxation

Write a linear program to solve set cover (and its fractional relaxation).

Fractional relaxation

Write a linear program to solve set cover (and its fractional relaxation).

Fractional relaxation

- The (fractional) optimal solution of the relaxation is a lower bound of the optimal solution of the original integer linear program.
- Example in which a fractional set cover may be cheaper than the optimal integral set cover:
  Input: $U = \{e, f, g\}$ and the specified sets $S_1 = \{e, f\}$, $S_2 = \{f, g\}$, $S_3 = \{e, g\}$, each of unit cost.
  - An integral cover of cost
  - A fractional cover of cost

Fractional relaxation

- The (fractional) optimal solution of the relaxation is a lower bound of the optimal solution of the original integer linear program.
- Example in which a fractional set cover may be cheaper than the optimal integral set cover:
  Input: $U = \{e, f, g\}$ and the specified sets $S_1 = \{e, f\}$, $S_2 = \{f, g\}$, $S_3 = \{e, g\}$, each of unit cost.
  - An integral cover of cost
  - A fractional cover of cost

A simple rounding algorithm

Algorithm:

- 1- Find an optimal solution to the LP-relaxation.
- 2- (Rounding) Pick all sets S for which $x_S \geq 1/f$ in this solution.

- Theorem: The algorithm achieves an approximation factor of f for the set cover problem.
- Proof: To be proved:
  - 1) All elements are covered.
  - 2) The cover returned by the algorithm is of cost at most $f \cdot OPT$

  Proofs:
  - proof of 1)
  - proof of 2)

- Theorem: The algorithm achieves an approximation factor of f for the set cover problem.
- Proof: To be proved:
  - 1) All elements are covered.
  - 2) The cover returned by the algorithm is of cost at most $f \cdot OPT$

Proofs:

- proof of 1)
- proof of 2)

- Theorem: The algorithm achieves an approximation factor of f for the set cover problem.
- Proof: To be proved:
  - 1) All elements are covered.
  - 2) The cover returned by the algorithm is of cost at most $f \cdot OPT$

Proofs:

- proof of 1)
- proof of 2)

- Theorem: The algorithm achieves an approximation factor of f for the set cover problem.
- Proof: To be proved:
  - 1) All elements are covered.
  - 2) The cover returned by the algorithm is of cost at most $f \cdot OPT$

Proofs:

- proof of 1)
- proof of 2)

Randomized rounding

- Idea: View the optimal fractional solutions as probabilities.
- Algorithm:
  - Flip coins with biases and round accordingly (S is in the cover with probability x_S).
  - Repeat the rouding $O(\log n)$ times.
- This leads to an $O(\log n)$ factor randomized approximation algorithm. That is
  - The set is covered with high probability.
  - The cover has expected cost: $O(\log n)OPT$.

Take Aways

- Fractional relaxation is a method to obtain for some problems:
  - Lower bounds on the optimal solution of an integer linear program (minimization).
    Remark: Used in Branch & Bound algorithms to cut branches.
  - Polynomial approximation algorithms (with rounding).
- Complexity:
  - Integer linear programs are often hard.
  - (Fractional) linear programs are quicker to solve (polynomial time).

Linear Programming
Frédéric Giroire   Year

TP n°

Linear Program Solvers

In this class, we learn how to solve a linear program on a computer using a solver (here GLPK). Then, we use the graph and linear program libraries of SAGEMATH to solve some combinatorial problems and networking problems. We finish by modelling a research problem : In a telecom backbone network, find a routing of the demands that minimizes the energy consumption of the network.

SAGEMATH is based on the PYTHON language. A crash course of PYTHON and a list on the functions to be used is given in this document.

Installation instructions and detailed documentation can be found :
- GLPK `http://www.glpk.fr`
- SAGEMATH `http://www.sagemath.org/index.html` The software can be used *online, without installation*. But you will need to create an account.

1 Using GLPK to solve Linear Programmes

We examine here the *input file format* of GLPK, its *commands* and *output format*. We then use it to solve a maximum matching problem.

1.1 Introduction to GLPK

1.1.1 File format.

Different formats exist. We use here CPLEX format which is widely used. Example :

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

The type of a variable is a *non-negative real number by defaut*. It can be set to *Integer* or *Binary* with the corresponding keywords, and to a possibly negative variable with the keyword *free*.

1.1.2 GLPK commands.

Do not hesitate to use `glpsol --help` to have a list and explanation of the program commands.

To launch the solver and solve the LP defined in the file prog.lp in the CPLEX file format, use : `glpsol --cpxlp prog.lp -o output.txt` The output is written in the file `output.txt`.

1.1.3 Output format.

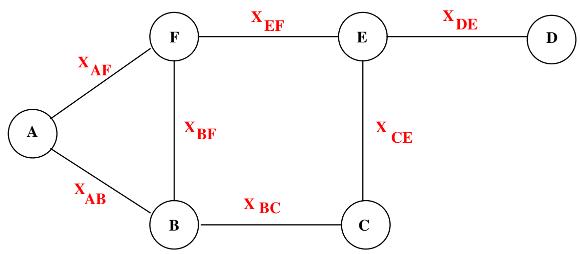
```
Problem:
Rows:       3
Columns:    4 (2 integer, 1 binary)
Non-zeros:  9
Status:     INTEGER OPTIMAL
Objective:  obj = 87.5 (MAXimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 c1                        -19                          20
     2 c2                         30                          30
     3 c3                          0             0             =

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 x1                       60.5
     2 x2                       10.5             0            40
     3 x3           *              1             0             1
     4 x4           *              3             2             3

Integer feasibility conditions:

KKT.PE: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

End of output
```

Karush-Kuhn-Tucker (KKT) optimality conditions are used to assess the numerical accuracy of the resulting solution.

1.2 Solving a Graph Problem with GLPK

Again, we look at the MAXIMUM MATCHING PROBLEM. A small reminder.

Definition. Let $G = (V, E)$ be a graph.
— A *matching* $M \subseteq E$ is a collection of edges such that every vertex of V is incident to at most one edge of M.
— The *maximum cardinality matching* problem is to find a matching M of maximum size.
On the following graph,



The problem can be modeled with the following LP :

$$\begin{array}{ll}
\text{Var. :} & x_{AB} = 1 \text{ if } AB \in M, \\
 & x_{AB} = 0 \text{ otherwise} \\
 & \\
\max & x_{AB} + x_{BC} + x_{CE} \\
 & +x_{DE} + x_{EF} + x_{AF} + x_{BF} \\
\text{s.t.} & \\
 & x_{AB} + x_{AF} \leq 1 \\
 & x_{AB} + x_{BC} + x_{BF} \leq 1 \\
 & x_{BC} + x_{CE} \leq 1 \\
 & x_{DE} \leq 1 \\
 & x_{CE} + x_{EF} + x_{DE} \leq 1 \\
 & x_{BF} + x_{EF} + x_{AF} \leq 1 \\
 & x_{AB}, x_{BC}, x_{CE}, x_{DE}, x_{EF}, x_{AF}, x_{BF} \geq 0 \\
 & x_{AB}, x_{BC}, x_{CE}, x_{DE}, x_{EF}, x_{AF}, x_{BF} \in \mathbb{N}
\end{array}$$

This LP is easily translated into a format readable by GLPK :

```
Maximize
  xAB+xBC+xCE+xDE+xEF+xAF+xBF
Subject to
  c1: xAB + xAF <= 1
  c2: xAB + xBC + xBF <= 1
  c3: xBC + xCE <= 1
  c4: xDE <= 1
  c5: xCE + xEF + xDE <= 1
  c6: xBF + xEF + xAF <= 1
Binary:
  xAB, xBC, xCE, xDE, xEF, xAF, xBF
```

The output given by GLPK is

```
Problem:
Rows:       6
Columns:    13 (7 integer, 7 binary)
Non-zeros:  14
Status:     INTEGER OPTIMAL
Objective:  obj = 3 (MAXimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 c1                          1                           1
     2 c2                          1                           1
```

```
     3 c3                          1                             1
     4 c4                          1                             1
     5 c5                          1                             1
     6 c6                          1                             1

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 xAB                         0             0
     2 xBC                         1             0
     3 xCE                         0             0
     4 xDE                         1             0
     5 xEF                         0             0
     6 xAF                         1             0
     7 xBF          *              0             0             1

Integer feasibility conditions:

INT.PE: max.abs.err. = 0.00e+00 on row 0
        max.rel.err. = 0.00e+00 on row 0
        High quality

INT.PB: max.abs.err. = 0.00e+00 on row 0
        max.rel.err. = 0.00e+00 on row 0
        High quality

End of output
```

The maximum matching is of size 3. The three edges of the matching are BC, DE, and AF.

2 Using Sagemath

Concise Formating

As we have seen in the former lessons, graph problems can be written in a more concise form and for a general graph. For example, here is a formulation of the Minimum Vertex Cover Problem : as :

$$
\begin{array}{lll}
\text{Var. :} & x_i = 1 \text{ if } i \in C, & \\
 & x_i = 0 \text{ otherwise} & \\
 & & \\
\min & \sum_{i \in V} x_i & \\
 & & \\
\text{s. t.} & & \\
 & x_i + x_j \geq 1 & (\forall ij \in E) \\
 & & \\
 & x_i \in \{0,1\} & (\forall i \in V)
\end{array}
$$

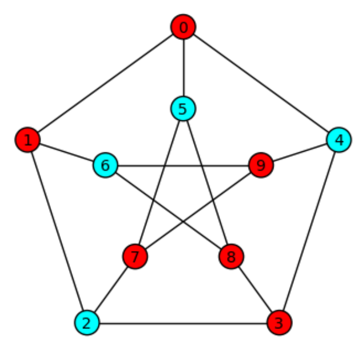
For the solver, the user has to write a small script in his language of preference to obtain the same generality. The script takes a graph as an input and outputs the LP files that will be the input of GLPK.

Here, to avoid have to script input-output we will use the graph library of the Sage software.

Solving a graph problem with Sage

Here is how to write the Minimum Vertex Cover Problem using *Sage*. The program takes the Petersen graph as an input.

```
# Define a graph
  sage: g=graphs.PetersenGraph()

# Define the linear program as a minimization problem
  sage: p=MixedIntegerLinearProgram(maximization=False)

# Defining the variables
 sage: b=p.new_variable(binary=True)

# Setting the constraints
  sage: for (u,v) in g.edges(labels=None):
  ...     p.add_constraint(b[u]+b[v] >= 1)

# Setting the objective function
  sage: p.set_objective(sum([b[u] for u in g]))

# Write the LP in a file under the LP format.
  sage: p.write_lp("max-matching-sage-lp.lp")

# Solving the linear program
  sage: p.solve()

# Printing the solution
  sage: b = p.get_values(b)
  sage: m = [u for u in g if b[u] == 1]
  sage: print m
  [0, 1, 3, 7, 8, 9]

# Drawing the solution
  sage: g.show(vertex_colors={"red":m})
```



3 Crash course of Python

Python can be seen as a very powerful scripting language allowing to write very quickly programs. In particular, it is a convenient language for class : all commands can be tested in a terminal as the language is interpreted (and the compilation is not necessary) ; the documentation is incorporated inside the terminal. You can get the doc of a function or object by writing its name followed by `?` and the source code with `??`. Additionaly, the list of methods of an object can be obtained by putting a dot after an object and pressing *tab*).

```
sage: g ?
Type:           Graph
Base Class:     <class 'sage.graphs.graph.Graph'>
String Form:    Graph on 0 vertices
Namespace:      Interactive
Length:         0
File:           /Applications/sage/local/lib/python2.6/site-packages/sage/graphs/graph.py
Docstring:
       Undirected graph.

       A graph is a set of vertices connected by edges. See also the
       Wikipedia article on graphs.

       One can very easily create a graph in Sage by typing:

          sage: g = Graph()
...

sage: g.
Display all 256 possibilities? (y or n)
g.add_cycle                             g.is_directed
g.add_edge                              g.is_drawn_free_of_edge_crossings
g.add_edges                             g.is_equitable
g.add_path                              g.is_eulerian
g.add_vertex                            g.is_even_hole_free
...
```

Below are presented quickly differences with other programming languages like C or Java, common commands, and nice tricks existing in this language.

Variables in Python are directly used without declaration.

```
x=10
name="frederic"
```

Contrary to language like C or Java, blocks are defined by indentation (and not by brackets {}). The indentation is done with a *tab* or a succession of 3 *spaces*. For example, a conditional statement would be defined in the following way in Python :

```
if x > 5 and x/2==0:
   print "OK"
```

Functions are defined by using the keyword `def`.

```
def sum(x,y):
   value = x+y
   return value
```

Lists are handled in a very natural way in Python. Defining a list :

```
l = [2,10,5]
```

Iterating on the elements of a list :

```
for x in l:
   print x
```

There is a very elegant functionality in Python called *list comprehension*. This is the possibility of creating a new list from a list.

```
l2 = [2*x for x in l if x > 4]
```

This command creates a list with the elements of l that are greater than 4.
A nice trick with tuples

```
x = (3,"2e element")
l = (x,(4,"et oui"))
for (a,b) in l:
   print "The value of ",b,"is ",a"."
```

Python commands and Sagemath functions for the exercises

The script below :
- Defines a graph g with 5 vertices, using constructor `Graph(n)`.
- Iterates on the vertices of the graphs and print their names. Note that, by default, nodes are designated by numbers.
- Adds an edge between vertices 1 and 2 with label 4, add and edge between vertices 3 and 4 with label 7, using method `g.add_edge(u,v,w)`.
- Iterates on the edges, using method `g.edges()`. Note that an edge is a triple, and that the values of its elements can be retrieved very elegantly. If you would like to retrieve only the end vertices, use `g.edges(labels=None)`.
- Retrieve the neighbors of vertex 1, using the method `g.neighbors(u)`.

```
sage: g = Graph(5)
sage: for v in g:
....:     print v
....:
0
1
2
3
4
sage: g.add_edge(1,2,4)    # an edge without weight can be added simply by g.add_edge(1,2)
sage: g.add_edge(3,4,7)
sage: for (u,v,w) in g.edges():
....:     print "Label of edge ",u,v,": ",w
....:
Label of edge  1 2 :  4
Label of edge  3 4 :  7
sage: g.neighbors(1)
[2]
```

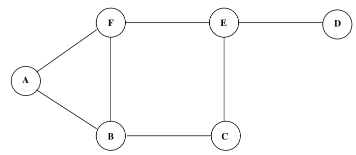
- Defining a digraph with 7 vertices, using constructor `DiGraph(n)`.
- To retrieve the in-neighbors and out-neighbors of a vertex, use the function `d.neighbors_in(v)` and `d.neighbors_out(v)`.

```
sage: d = DiGraph(7)
sage: d.add_edge(3,4,7)
sage: d.add_edge(3,1,8)
sage: d.neighbors_out(3)
[1, 4]
```

4 Exercises

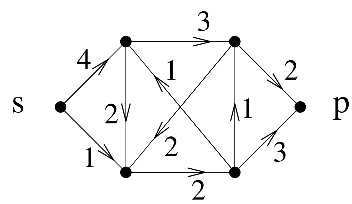
4.1 Solving linear programs using GLPK

Exercise 1 Consider the VERTEX COVER PROBLEM on the following graph :



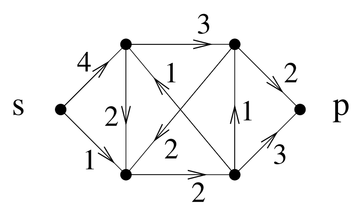
Solve the problem using GLPK.

Exercise 2 Consider the SHORTEST PATH PROBLEM on the following graph :



Solve the problem using GLPK.

Exercise 3 Consider the MAXIMUM FLOW PROBLEM on the following graph :



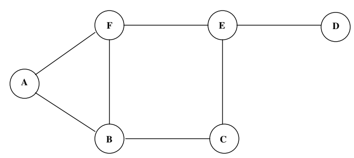
Solve the problem using GLPK.

4.2 Solving Graph and Network Problems using `Sage`

Exercise 4 [Maximum Independant Set with Sage] Write a Sage script to solve the MAXIMUM INDEPENDANT SET PROBLEM on the Petersen graph.

Exercise 5 [Maximum Matching with Sage]

1. Write a Sage script that solves the MAXIMUM MATCHING on the following graph (use a function taking a graph as a parameter `maximumMatching(g)`, define the graph below and call the function)

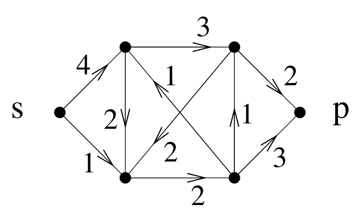


   Hint : Beware not to use both variables b[(u,v)] and b[(v,u)], as there is only a single edge uv. You may use the following trick : You define the function `B`, *as* `B = lambda x, y : b[ (x,y) if x<y else (y,x)]`. *You then call* `B(u,v)` *instead of* `b[(u,v)]`.

2. Relax the integer property of the variables and relaunch the program on the above graph. What is the solution and its value ? Comment.

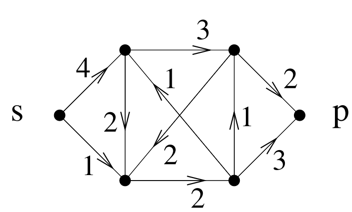
Exercise 6 [Shortest Path with Sage]

Write a Sage script that solves the SHORTEST PATH PROBLEM from the node s to the node p on the following weighted graph and plots the shortest path in red on the digraph. (use a function taking a directed graph as a parameter `shortestPath(d)`, define the digraph below with labels on the arcs, and call the function).



Exercise 7 [Flows with Sage*]

1. Write a Sage script that solves the MAXIMUM FLOW PROBLEM from the node s to the node p on the following network (use a function taking a directed graph as a parameter `maximumFlow(d)`, define the digraph below with labels on the arcs for the capacities, and call the function). Plots the flows and capacities on the arcs of the digraph.



2. There exists a second classic linear formulation of the MAXIMUM FLOW PROBLEM, the *path-formulation*.

We note $\mathcal{P}$ the set of all paths going from s to t. In this formulation, there is a variable $f_p \in \mathbb{R}^+$ per path $p \in \mathcal{P}$ giving the value of the flow on path p. The Maximum Flow Problem can be written as :

$$\begin{array}{lll} \max & \sum_{p \in \mathcal{P}} f_p & \\ \text{s. t.} & & \\ & \sum_{ij \in A} \leq c_{ij} & (\forall ij \in A) \\ & f_p \in \mathbb{R}^+ & (\forall p \in \mathcal{P}) \end{array}$$

Write a second Sage function solving the Maximum Flow Problem using this formulation. You may use the Sage function `d.all_simple_paths()`.

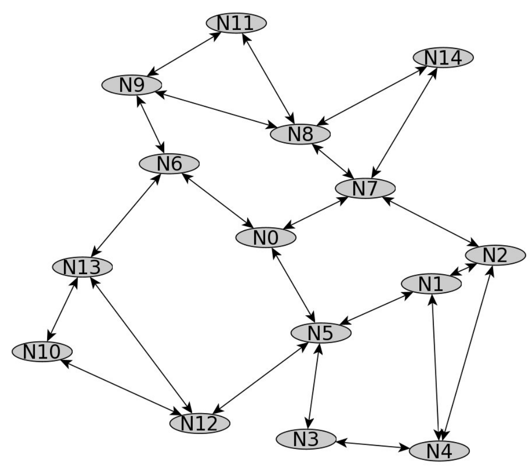
3. Launch both functions on complete digraphs of sizes 3,4,5, ... Compare the implementation time. Comment.
4. A Telecom company wants to send a maximum flow on the above network, but it wants to transport flow only on paths with less than 4 edges, as longer paths imply longer delays. Which formulation the company should use ? Comment.

Exercise 8 [Multicommodity Flows with Sage]

1. Write a Sage script to solve the Multicommodity Flow Problem on the Petersen graph for an all-to-all demand. You should define a function taking as parameter a network and a demand matrix.
2. Consider a variant of the Multicommodity Flow Problem in which a flow should be an integer and should use only one path from its source to its destination. Write the corresponding LP. Launch it on the Petersen graph. Comment (nature of the solution, execution time).

4.3 Modelling and Solving a Research Problem

Exercise 9 [Reducing Network Energy Consumption *] A backbone network is a core network of today's Internet. It is operated by a large company such as France Telecom in France. Schematically, it is made of computers routing the traffic called routers and of optical fibers linking them. Its role is to transfer the Internet traffic from a big city to an other big city. It can be modelled as a network, that is a digraph $G = (V, E)$ with a weight function $c : E \to \mathbb{R}$ and a demand matrix D, where D_{ij} is the traffic demand from city i to city j. Each bi-directional link uv corresponds to two arcs $u \to v$ and $v \to u$. The capacity of all arcs are equal and of value 1. An example of such networks is given below :



We consider here the problem of reducing backbone network energy consumption. Recent studies have shown that the energy consumption of routers does not depend of the traffic load that they route, but mainly on the number of active links between them. We suppose here that links that are not used can be turned-off.

1. Model the problem of minimizing the backbone energy consumption while routing all the demands as a linear program (you may start from the program for the multicommodity flow provided by the professor).
2. We want to study 4-regular square grid networks that are often used by operators. Use the `Sage` software to find the routing using the minimum amount of energy for a 3 by 3 grid for an all-to-all traffic demand of intensity 1/10, that is $D_{ij} = 0.1$ for all $i, j \in V$. What is the percentage of energy that can be saved ?
3. The traffic of the operator is highly dynamic. For example, it varies during the day and is of low level at night. We want to implement an algorithm adapting to the level of trafic. To do so, we first want to assess the percentage of energy that can be saved for different levels of traffic.

   What is the maximum level traffic $D_{\max}$ that can be routed by a $n \times n$-grid (when $D_{ij} = D_{\max}, \forall i, j \in V$) ? Write a sage script that returns the energy saving for 5 different levels of traffic between 0 and $D_{\max}$ in a 3×3 grid.
4. Try to do the same study for 4×4-grids, 5×5-grids,... what happens ?
5. Propose solutions. In particular, propose a heuristic algorithm able to provide solutions for large networks.
6. *Virtual network functions.* Now each flow is divided into 2 subflows of equal size. The first subflow has to pass through 2 network functions, a firewall and a DPI (Deep Packet

Inspection). The second one, through 3 network functions : a firewall, a TCP optimizer and a Video Optimizer. Each node has capacity 2, meaning that each node can host at most 2 virtual network functions.
Write an ILP and then a script solving the energy aware routing while doing the assigment of network functions. You will plot the positions on network functions in the network. Solve first the problem with only 1 flow, then 2 flows, ... For how many flows, you may solve the problem ?

Exercise 10 [Reducing Network Energy Consumption using Virtualization *]
Network Function Virtualization (NFV) is a promising network architecture concept to reduce operational costs. In legacy networks, network functions, such as firewall or TCP optimization, are performed by specific hardware. In networks enabling NFV coupled with the Software Defined Network (SDN) paradigm, Virtual Network Functions (VNFs) can be implemented dynamically on generic hardware. This is of primary interest to implement energy efficient solutions, in order to adapt the resource usage dynamically to the demand. In this project, we *study how to use NFV coupled with SDN to improve the energy efficiency of networks.*

1. Model the problem of minimizing the backbone energy consumption while routing all the demands as a linear program (you may start from the program for the multicommodity flow provided by the professor).
2. Use Sagemath and the glpk solver to solve the problem on larger and larger grids. What happens ?
2. Propose a heuristic algorithm to find a good solution of the problem for large networks.
3. Solve the problem using the heuristic algorithm. Compare the solutions of the ILP with the one of the heuristic algorithm (energy saved and execution time).

We consider now a setting in which a flow has to go through a Service Function Chain, that is several network functions in a specific order.

4. Propose an ILP to solve this variant of the problem.
5. Propose a heuristic algorithm to solve this variant of the problem.
6. Compare the solutions provided by the ILP and the algorithm in terms of energy saved and execution time.