Contents

1 Linear Programming (LP)

LP problems originally appeared in Operations Research. The form of an LP problem is as follows:

$$\text{minimize } \mathbf{c}^T\mathbf{x}, \tag{1}$$

subject to the constraints:

$$A\mathbf{x} = \mathbf{b} \text{ or } A\mathbf{x} \geq \mathbf{b}$$

$$\mathbf{x} \geq \mathbf{0}, \tag{2}$$

with $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{R}^n$, $\mathbf{c} = (c_1, c_2, \ldots, c_n) \in \mathcal{R}^n$, $\mathbf{b} = (b_1, b_2, \ldots, b_m) \in \mathcal{R}^m$.

Function $\mathbf{c}^T\mathbf{x} : \mathcal{R}^n \to R$ is called the *objective function* and $A\mathbf{x} \leq \mathbf{b}$ are called constraints. More specifically:

- c_i is the cost per unit of variable x_i.
- The total cost can be represented by $\mathbf{c}^T\mathbf{x} = c_1x_1 + \ldots + c_nx_n$.
- x_i is the i-th variable i.

The constraints and the objective function are linear to vector of variables $\mathbf{x}$. Matrix $A \in \mathcal{R}^{mxn}$ is a $m \times n$ matrix,

$\mathbf{b} \in \mathcal{R}^m$, and

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

Therefore the problem can be written as:

$$\text{minimize} \;\; c_1x_1 + \cdots + c_nx_n$$

$$\text{s.t.} \quad a_{11}x_1 + \cdots + a_{1n}x_n = b_1$$

$$\vdots$$

$$a_{m1}x_1 + \cdots + a_{mn}x_n = b_m$$

or

$$\text{minimize} \;\; c_1x_1 + \cdots + c_nx_n,$$

$$\mathbf{a}_i^T\mathbf{x} = b_i, \tag{3}$$

where $\mathbf{a}_i$ is the i-th row of matrix A, for $i = 1, \ldots, m$. We will focus on formulating and solving LP problems.

1.1 Example 1 : A company with 4 products

A company constructs four products : $\Pi_1, \Pi_2, \Pi_3, \Pi_4$.The resources that are needed are: man-weeks, kg of material A and quantity of material B (in packages).

Resources	Π_1	Π_2	Π_3	Π_4	Resources
man-weeks	1	2	1	2	20
kg of material A	6	5	3	2	100
packages of material B	3	4	9	12	75

Each cell (i, j) in the table above contains the number of units of resource i which are necessary to produce one unit of product j. Thus, for example Π_2,Π_4 are the most demanding ones in man-weeks. Also, 6 kilograms of material A are needed to make one unit of Π_1.

The last column of the table shown the availability of resources. Availability shows the amounts of the resources that the company can waste to produce the products. So availability is going to be vector $\mathbf{b}$ with $m = 3$ elements and the table is going to be matrix A of the Linear Program.

There is also a cost vector $[6\ 4\ 7\ 5]$, where each cost coefficient c_i expresses the benefit of the company for each unit of product $\Pi_i, i = 1, 2, 3, 4$ that is sold. Thus $c_1 = 6$ is the profit per unit of product Π_1.

The company's objective is to find the vector
$\mathbf{x} = (x_1, x_2, x_3, x_4)$ with x_i the quantity of Π_i that must be

constructed so as to maximize the total benefit from all products. We assume that vector $x \in R^n$. Note that if $x \in Z^n$ then the problem would be an *integer programming problem* which are computational problems not solvable in polynomial time.
The problem is stated as follows:

$$\text{maximize } \mathbf{c}^T\mathbf{x} = 6x_1 + 4x_2 + 7x_3 + 5x_4 \tag{4}$$

subject to the constraints :

$$x_1 + 2x_2 + x_3 + 2x_4 = 20 \tag{5}$$

$$6x_1 + 5x_2 + 3x_3 + 2x_4 = 100 \tag{6}$$

$$3x_1 + 4x_2 + 9x_3 + 12x_4 = 75 \tag{7}$$

and $\mathbf{x} = (x_1, x_2, x_3, x_4) \geq \mathbf{0}$.

In the formulation, we assumed that all available resources are used.

1.2 Example 2 : Diet Problem

There are n different kinds of food and m vitamins. Each unit of food j costs c_j, $j = 1, \ldots, n$. To achieve balanced diet, we must receive at least b_i units of vitamin i per day, $i = 1, \ldots, m$.

A unit of food j contains a_{ij} units of vitamin i. Elements a_{ij} form matrix A . $\mathbf{x} = (x_1, \ldots, x_n)$ is the vector of variables, where x_j is the amount of food j in the diet. We want to find the quality x_j of each food j that should be consumed per day, so that all necessary vitamins are received and the cost is minimized. This is the min-cost diet problem which can be formulated as follows:

$$\text{minimize } \sum_{i=1}^{n} c_i x_i = \mathbf{c}^T \mathbf{x}$$

subject to

$$a_{11}x_1 + \cdots + a_{1n}x_n \geq b_1$$

$$\vdots$$

$$a_{m1}x_1 + \cdots + a_{mn}x_n \geq b_m$$

and $\mathbf{x} = (x_1, \ldots, x_n) \geq (0, 0, \ldots, 0)$ or,

$$\min \quad \mathbf{c}^T \mathbf{x} \tag{8}$$

subject to:

$$A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}. \tag{9}$$

2 Geometrical interpretation of LP Problems

Consider the problem:

$$\max \begin{pmatrix} 1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\text{s.t.} \quad \begin{pmatrix} 5 & 6 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 30 \\ 12 \end{pmatrix} \text{ with } x_1, x_2 \geq 0.$$

Infinitely many points (x_1, x_2) satisfy the two constraints and the set of feasible solutions is all such points. Now, we draw the region of feasible solutions. This is shaded area OABC in the figure below.

Later we will see that the optimal solution $\mathbf{x}^* = (x_1^*, x_2^*)$, i.e. the one that maximizes the objective $x_1 + 5x_2$ is *always* one of the four vertices O,A,B or C.

Geometrically, maximizing $\mathbf{c}^T\mathbf{x} = x_1 + 5x_2$ subject to $\mathbf{x} \in (OABC)$ amounts to finding a straight line $x_1 + 5x_2 = a$ that intersects with the shaded region and has the largest value, a.

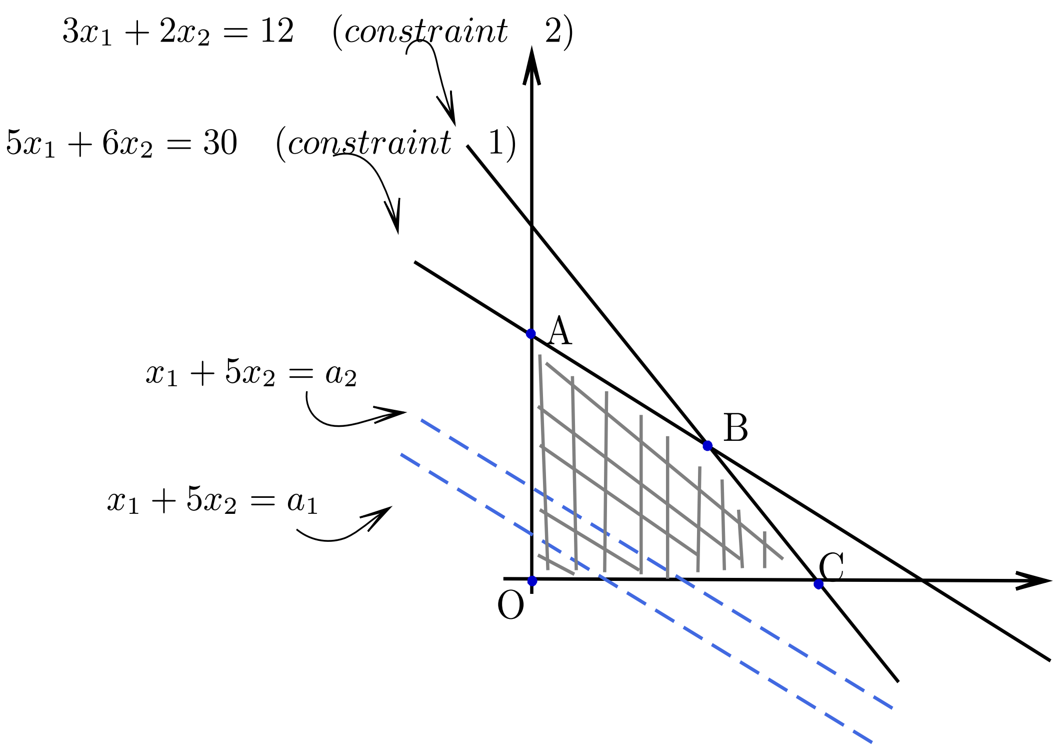

Figure 1: The feasible region of an LP problem.

Thus, we can draw the lines $x_1 + 5x_2 = a$ and consecutively increase a to values $a_1 < a_2 < \ldots$. Thus, we form the parallel lines

$$x_1 + 5x_2 = a_1$$
$$x_1 + 5x_2 = a_2$$
$$\vdots$$
$$x_1 + 5x_2 = a_{\text{max}},$$

until we reach the value a_{max}, beyond which if we increase a further, we will go out of the feasible region. Then a_{max} is the maximum value of the objective function, and the point of intersection of $x_1 + 5x_2 = a_{\text{max}}$ with region (OABC) is the optimal solution.

In general the LP problem is of the form:

$$\text{minimize } \mathbf{c}^T\mathbf{x} \tag{10}$$

$$\text{s.t.} \quad \mathbf{x} \in \mathcal{P}. \tag{11}$$

Then $\mathcal{P}$ is called set of feasible solutions of the LP problem and is a polyhedron.

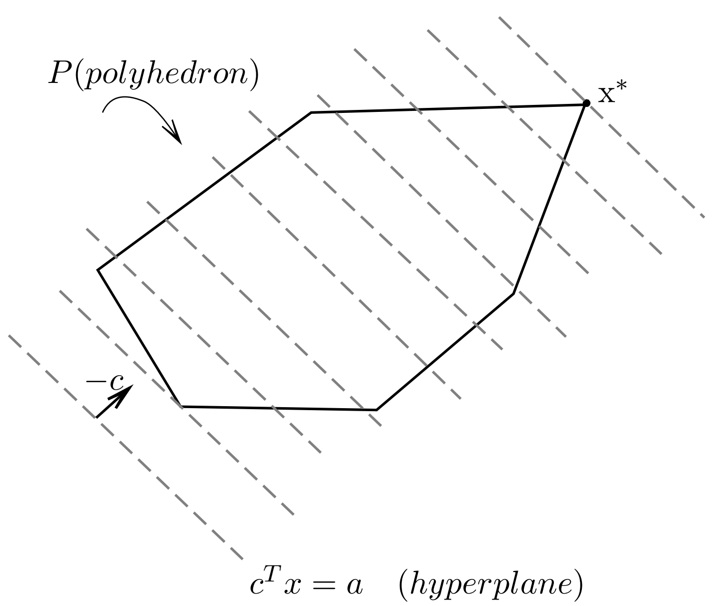

Figure 2: Feasible region of LP problems is a polyhedron.

Objective $\mathbf{c}^T\mathbf{x}$ is linear in the vector of variables $\mathbf{x}$, so its level curves are *hyperplanes* orthogonal to $\mathbf{c}$ (shown by dashed lines).

The optimal solution $\mathbf{x}^*$ is the point in $\mathcal{P}$ as far as possible in direction $-\mathbf{c}$. Sometimes the optimal solution is not only one point but several.
Also, it is important to note that the optimal solution of a LP problem is always one of the vertices of the corresponding polyhedron. For example, if we have a LP problem of two variables and an optimal solution for this problem exists then this will be one of the vertices of the corresponding quadrilateral.

2.1 Cases of an LP problem solution

Consider the LP problem

$$\text{minimize } c_1x_1 + c_2x_2 \tag{12}$$

subject to:

$$-x_1+x_2 \leq 1, \text{ with } x_1 \geq 0, \ , x_2 \geq 0, \ A = [-1 \ 1] \ b = [1]. \tag{13}$$

We have the following cases with regard to a solution:

1. An LP problem may have a unique solution, e.g. when $\mathbf{c} = (1, 1)$, then $\Rightarrow x_1^* = 0, x_2^* = 0 \Rightarrow \mathbf{x}^* = [0, \ 0]$ is the unique optimal solution.

2. The problem may have multiple optimal solutions.

   – If $\mathbf{c} = (1, 0)$, then any vector $(0, x_2)$ is optimal with $x_2 \in [0, 1]$. The set of the optimal solutions is infinite but bounded, since $0 \leq x_2 \leq 1$.

   – If $\mathbf{c} = (0, 1)$, then there exist several optimal solutions of the form$(x_1, 0)$ with $x_1 \in [0, \infty]$. The set of the optimal solutions is infinite and unbounded in this case.

   We can easily understand that, that is because the variable x_1 and x_2 in each case respectively does not participate in the computation of total cost.

3. An LP problem has optimal cost $-\infty$ and no finite feasible solution. For example, if $\mathbf{c} = (-1, -1)$, then for the problem

$$
\begin{aligned}
&\text{min } (-x_1 - x_2) \\
&\text{s.t. } x_2 \leq 1 + x_1
\end{aligned}
$$

   for any feasible solution (x_1, x_2), we can produce another feasible solution with less cost by simply increasing x_1. By considering vectors with increasing values of x_1, x_2, we obtain a sequence of feasible solutions that goes to $-\infty$.

3 Standard form of an LP Problem

An LP problem is said to be in standard form if it is of the form:

$$\text{min } \mathbf{c}^T\mathbf{x}$$
$$\text{s.t. } A\mathbf{x} = \mathbf{b} \text{ with } \mathbf{x} \geq \mathbf{0}$$
$$A \in \mathcal{R}^{m\times n},\ m < n,\ \text{rank}(A) = m,\ \mathbf{b} \geq \mathbf{0}.$$

Namely, an LP is said to be in *standard* form has equality constraints, and is a minimization problem.

An LP problem is in *inequality* form if it is of the form:

$$\text{min } \mathbf{c}^T\mathbf{x}$$
$$\text{s.t. } A\mathbf{x} \geq \mathbf{b} \text{ with } \mathbf{x} \geq \mathbf{0}$$
$$A \in \mathcal{R}^{m\times n},\ m < n,\ \text{rank}(A) = m,\ \mathbf{b} \geq \mathbf{0}.$$

Those two forms are equivalent in the sense that starting from a feasible solution of a standard form problem we can produce a feasible solution of an inequality form problem with the same cost (and vice versa):

Standard form		Inequality form
feasible(optimal)	$\Leftrightarrow$	feasible(optimal)

Consider a non-standard LP problem:

$$\text{min } \mathbf{c}^T\mathbf{x}$$
$$\text{s.t. } A\mathbf{x} \geq \mathbf{b} \text{ with } \mathbf{x} \geq \mathbf{0}.$$

In order to convert it to standard form, we need to convert the inequalities to equalities. We subtract a positive quantity y_i out of each constraint i. We call $y_i, i = 1, 2 \ldots, m$ *surplus variables*, with $\mathbf{y} \geq \mathbf{0}$. Then, we have:

$$
\begin{gathered}
a_{11}x_1 + \ldots + a_{1n}x_n \geq b_1 \;\; \Rightarrow \\
a_{11}x_1 + \ldots + a_{1n}x_n - y_m = b_1 \\
\vdots \\
a_{m1}x_1 + \ldots + a_{mn}x_n \geq b_m \;\; \Rightarrow \\
a_{m1}x_1 + \ldots + a_{mn}x_n - y_m = b_m
\end{gathered}
$$

or in matrix form it is written as: $(A \quad -I_m) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{b}$ where $(A \quad -I_m)$ is a block matrix and I_m is the $m \times m$ unit matrix. Thus, the non-standard LP problem is transformed into a standard LP problem:

$$
\begin{gathered}
\text{min } \mathbf{c}^T\mathbf{x} \\
\text{s.t.} \\
A\mathbf{x} - \mathbf{y} = [A \quad -I_m] \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{b} \qquad \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}.
\end{gathered}
$$

Note that the vector of variables is $(\mathbf{x}, \mathbf{y})$ but the cost is the same as above, $\mathbf{c}^T\mathbf{x} + \mathbf{0}^T\mathbf{y} = \mathbf{c}^T\mathbf{x}$.

If the problem is in the non-standard form:

$$
\begin{gathered}
\text{min } \mathbf{c}^T\mathbf{x} \\
\text{s.t. } A\mathbf{x} \leq \mathbf{b} \text{ with } \mathbf{x} \geq \mathbf{0},
\end{gathered}
$$

we need to define positive variables $y_i, i = 1, 2 \ldots, m$, to add to each constraint, which we call *slack variables* and $\mathbf{y} \geq \mathbf{0}$. The new form of the problem is:

$$\min \mathbf{c}^T \mathbf{x}$$
$$\text{s.t. } A\mathbf{x} + \mathbf{y} = [A \quad I_m] \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{b} \qquad \mathbf{x} \geq \mathbf{0},\ \mathbf{y} \geq \mathbf{0}.$$

Note: All algorithms that we will examine later work for LP problems in *Standard form* only.

3.1 Exercise

We start from the problem:

$$\begin{array}{ll} \textit{max} & x_2 - x_1 \\ \textit{s.t.} & 3x_1 = x_2 - 5 \\ & |x_2| \leq 2 \\ & x_1 \leq 0 \end{array}$$

In order to bring an LP problem to a standard form, we change "max" to "min" and we need to transform all variables to non-negative ones. Also we transform all inequalities in the problem into equalities. So, the variable x_1 will be replaced by the variable $x_1' = -x_1$ and the two inequalities implied by $|x_2| \leq 2$ will be converted to equalities using slack variables x_3 and x_4. Now the problem is expressed as:

$$\begin{array}{ll} \textit{min} & -x_2 - x_1' \\ \textit{s.t.} & -3x_1' = x_2 - 5 \\ & x_2 + x_3 = 2,\ -2 + x_4 = x_2 \\ & x_3, x_4, x_1' \geq 0 \end{array}$$

In the original form of the problem, we have inequality $-2 \leq x_2 \leq 2$, x_2 should be redefined as $u - v$ with $u, v \geq 0$. The reason is that a variable which is unrestricted in sign (such as x_2) can be written in general as the difference of two positive variables. So in its standard form, the problem above becomes:

$$\begin{array}{ll} \textit{min} & -x_2 - x_1' \\ \textit{s.t.} & 3x_1' = 5 - x_2 \\ & u - v + x_3 = 2,\ v - u + x_4 = 2 \\ & x_3, x_4, x_1', v, u \geq 0 \end{array}$$

4 Routing as an LP problem

Routing in communication networks means selecting paths for transferring traffic from given source(s) to given destination(s).

Consider a network which is abstracted as a directed graph $G(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of edges of the graph. Let $\mathcal{N} = n$ be the number of nodes of the network. For each edge $(i, j) \in \mathcal{A}$ we define u_{ij} to be the *capacity* of edge junction (i, j), which is the maximum amount of traffic (in bps) that can be carried over the edge. Also, let c_{ij} be the cost per unit of transmitted traffic over the edge (i, j).

For each source k and destination l, define as b^{kl} the amount of traffic (bps) that is generated by node k and needs to be transferred to l. In this example, all nodes may be sources and destinations (if not, then $b^{kl} = 0$).

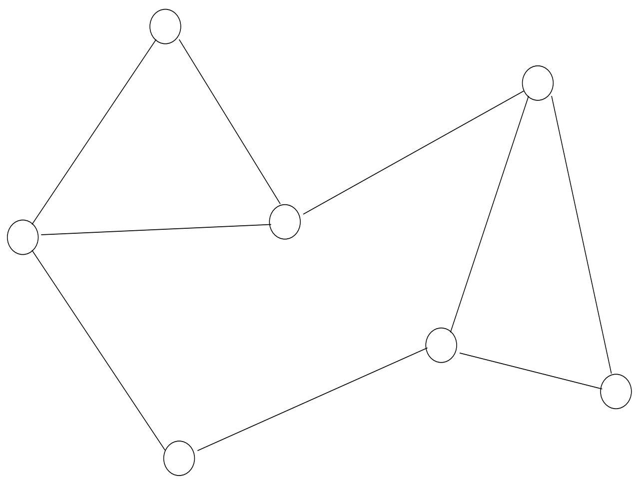

Figure 3: A graph, depicting a network (note here the graph is directed).

Problem:

Choose the paths for routing traffic for each source $k = 1 \ldots n$ to each destination $l = 1 \ldots n$ while minimizing total cost.

The variables in this problem are defined as a flow vector $f = \left(f_{ij}^{kl} : (i,j) \in A\right)$ where f_{ij}^{kl} is the amount of traffic with origin k and destination l that traverses link $(i,j) \in \mathcal{A}$.

For each node $i = 1 \ldots n$ we define:

$$b_i^{kl} = \begin{cases} b^{kl} & if \;\; i = k \\ -b^{kl} & if \;\; i = l \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$

Thus, b_i^{kl} denotes the net inflow at node i of traffic originated at k and destined at l.
Note that There are tree signicant kinds of routing:

- Single path routing: where each source node selects the shortest path to send data to its corresponding destination.
- Multipath routing: where load is devided evently among the first shortest paths in order to be send from the source to the destination.
- Mincost routing: where the data must be transmitted with the minimum overall cost. Note that Mincost routing can be either Singlepath or Multipath.

So, considering the *Mincost* routing, we have the problem:

$$\begin{aligned} \textit{min} \quad & \sum_{(i,j)\in\mathcal{A}} \sum_{k=1}^{n} \sum_{l=1}^{n} c_{ij}\, f_{ij}^{kl} \\ \textit{s.t.} \quad & 0 \leq \sum_{k=1}^{n} \sum_{l=1}^{n} f_{ij}^{kl} \leq u_{ij},\ \forall k,l = 1\ldots n \text{ and } \forall (i,j)\in\mathcal{A} \\ & \sum_{j:(i,j)\in\mathcal{A}} x_{ij}^{kl} - \sum_{j:(i,j)\in\mathcal{A}} f_{ji}^{kl} = b_i^{kl},\ \forall i = 1\ldots n \end{aligned}$$

In the problem there is a set of constraints, one for each node that reflect the *flow conservation constraint* at each node. Also, there is a constraint for each link that denotes the capacity constraint for each link.

The problem is called *minimum cost network flow problem*, and as we will see later, there are several known problems that emerge as special cases of this, such as the shortest path, the Max flow and the assignment problem.

5 BFS (Basic Feasible Solution)

Consider the system of inequalities $A\mathbf{x} = \mathbf{b}$, with $\mathbf{x} \geq 0$ and matrix A of dimension $m \times n$, $m \leq n$ and $rank(A) = m$. Matrix A can be written in a block matrix form as $A = [B \ \ D]$, where

(i) the $m \times m$ matrix B includes all m linearly independent

columns of A and
(ii) the $m \times (n - m)$ matrix D includes the rest of the columns of A.

By definition, B is non-singular ($|B| \neq 0$) where $|B|$ is the determinant of matrix B. Then, matrix B is said to be the *basis* for the system. The columns of B called *basic columns*. Then, the system of equations $B\mathbf{x}_B = \mathbf{b}$ has a unique solution, $\mathbf{x}_B = B^{-1}\mathbf{b}$. Vector $\mathbf{x}_B$ is of dimension $m \times 1$ and consists of those variables that correspond to the columns of B (these are called *basic variables*.

Thus, for example if A is 2×4 and the first and third column of A are linearly independent, then $\mathbf{x}_B = (x_1, x_3)^T$ and $x_2 = x_4 = 0$.

Note that vector $\mathbf{x} = (\mathbf{x}_B \;\; \mathbf{0})^T$ solves the original system $A\mathbf{x} = \mathbf{b}$ because:
$A\mathbf{x} = [B \;\; D] \; (\mathbf{x}_B \;\; \mathbf{0})^T = B \cdot \mathbf{x}_B + D \cdot 0 = \mathbf{b}$

5.1 Definitions

Definition 1: We call a vector of the form $(\mathbf{x}_B \;\; \mathbf{0})^T$ a *Basic solution* with respect to the basis B. Thus, all vectors of values variables that can be divided into a non-zero and a zero variable part (the non-zero part corresponding to the columns of a basis) are called *Basic solutions* (Note here that this definition does not imply feasibility).

Defintion 2: A basic solution $\mathbf{x} = (\mathbf{x}_B \;\; \mathbf{0})^T$ that satisfies $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ (i.e. it is feasible) is called *Basic Feasible Solution (BFS)*.

- If the BFS $\mathbf{x}_B > \mathbf{0}$ has all m components positive, the BFS is called *non-degenerate BFS*.
- Otherwise, if some of the components of $\mathbf{x}_B$ are zero (i.e the positive components of $\mathbf{x}_B$ are fewer than m) , the BFS is called *degenerate BFS*.

An alternative definition: Consider an LP problem with constraints $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$ and A a $m \times n$ matrix, $m \leq n$ (with m linear independent rows). Then $\mathbf{x} \in \mathcal{R}^n$ is a *Basic Feasible Solution* if there exist indices $B(1), ..., B(m)$ such that columns $\mathbf{a}_{B(1)}, \mathbf{a}_{B(2)}, \ldots, \mathbf{a}_{B(m)}$ of matrix A are linearly independent and $\forall i \neq B(1), B(2), ..., B(m)$ it is $x_i = 0$. Also it should be $A\mathbf{x} = \mathbf{b}$. In addition if $x_i > 0 \;\; \forall i \in \{B(1), B(2), ..., B(m)\} \Rightarrow \mathbf{x}$ is a non-degenerate BFS.

5.2 Example

Consider the set of constraints:

$$\begin{array}{ccccccl} x_1 & + & x_2 & + & 2x_3 & \leq 8 \\ & & x_2 & + & 6x_3 & \leq 12 \\ x_1 & & & & & \leq 4 \\ & & x_2 & & & \leq 6 \\ x_1 & , & x_2 & , & x_3 & \geq 0 \end{array}$$

After converting them to a standard form, we get:

$$\begin{array}{ccccccccccc} x_1 & + & x_2 & + & 2x_3 & + & x_4 & = 8 \\ & & x_2 & + & 6x_3 & + & x_5 & = 12 \\ x_1 & & & & & + & x_6 & = 4 \\ & & x_2 & & & + & x_7 & = 6 \\ x_1 & , & x_2 & , & \dots & , & x_7 & \geq 0 \end{array}$$

So the constraints correspond to the following representation: $A\mathbf{x} = \mathbf{b}$ where:
$\mathbf{x} = [x_1, x_2, \dots, x_7]^T$,
$\mathbf{b} = [8, 12, 4, 6]^T$,

$$A = \begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Let the basis be $B = [\, \mathbf{a}_1 \; \mathbf{a}_3 \; \mathbf{a}_4 \; \mathbf{a}_7 \,]$. If we consider the system of equations $B\mathbf{x}_B = \mathbf{b}$, the solution to that is $\mathbf{x}_B = [4, 2, 0, 6]^T$ and $\mathbf{x} = [4, 0, 2, 0, 0, 0, 6]^T$ is a BFS which is degenerate (since $x_4 = 0$).
Observe that in the case that we choose the basis to be $B = [\, \mathbf{a}_4 \; \mathbf{a}_5 \; \mathbf{a}_6 \; \mathbf{a}_7 \,]$ then we have solution $\mathbf{x}_B = [8, 12, 4, 6]^T$ and $\mathbf{x} = [0, 0, 0, 8, 12, 4, 6]^T$ is a BFS that is non-degenerate.

Clearly, there are several ways of choosing the basis for an

LP problem. For a matrix A of dimension $m \times n$, $m \leq n$ with $rank(A) = m$, we can choose among $C(n, m) = \frac{n!}{(n-m!)m!}$ different bases B /textbfat most, and so we can have a corresponding number of basic solutions.

5.3 Basic LP theorem

The basic theorem in LP is that in order to solve a LP problem, we will only need to check the BFS and among them find the optimal BFS, i.e the BFS that minimizes $\mathbf{c}^T\mathbf{x}$.

- (a) If there exists a feasible solution in an LP problem (i.e. in a 2-dimensional problem, the corresponding quadrilateral is not empty), then there exists a BFS.

- (b) If there exists an optimal feasible solution in an LP problem, then there exists an optimal BFS. Note that in an LP problem the optimal solution is always a BSF i.e. is of the form $\mathbf{x} = (\mathbf{x}_B \;\; \mathbf{0})^T$.

6 Fundamental theorem of Linear Programming

Before we present the above theorem, we will state some significant definitions.

Definition 1: Consider a polyhedron $\mathcal{P}$, defined by linear equality and inequality constraints as defined in previous sections. Then, the point (represented by vector) $\mathbf{x}_0$ is a *vertex* of $\mathcal{P}$ if there exists $\mathbf{c}$ such that $\mathbf{c}^T\mathbf{x}_0 < \mathbf{c}^T\mathbf{y}$ for all $\mathbf{y} \in P$ with $\mathbf{y} \neq \mathbf{x}_0$.

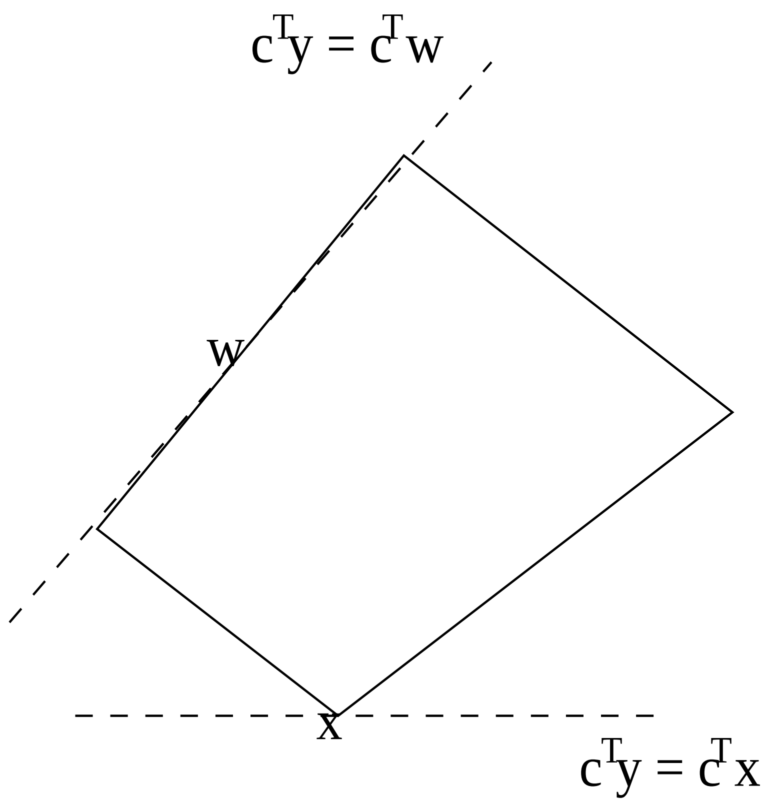

Figure 4: A vertex point $\mathbf{x}$ and a point $\mathbf{w}$ that is not a vertex.

As shown in figure 4 , $\mathbf{w}$ in the figure is not a vertex because there is no hyperplane that meets $\mathcal{P}$ only at $\mathbf{w}$. That is, a point $\mathbf{w}$ of a polyhedron is a vertex of the polyhedron if and only if a hyperplane passing through the point divides the space in two half-spaces and *all points of the hyperplane except from* $\mathbf{w}$ *lie on the same side (same half-space)*. In the example of figure 4 ,there exist all the points $\mathbf{y}$ on the hyper-plane $\mathbf{c}^T\mathbf{y} = \mathbf{c}^T\mathbf{w}$ that make the definition of $\mathbf{w}$ being vertex of $\mathcal{P}$ not hold.

Defintion 2: A point $\mathbf{x}$ is called *extreme point* of $\mathcal{P}$ if there are no distinct points $\mathbf{x}_1$,$\mathbf{x}_2$ of $\mathcal{P}$ such that $\mathbf{x} = \alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$. In other words, if $\mathbf{x}$ is an extreme point and it is $\mathbf{x} = \alpha\mathbf{x}_1 + (1 - \alpha\mathbf{x}_2)$ for some $\alpha \in (0, 1)$, then it must be $\mathbf{x} = \mathbf{x}_1 = \mathbf{x}_2$. Thus, an extreme point of a polyhedron cannot be represented as a convex combination of two other points of the polyhedron.

The Theorem (Fundamental theorem of Linear Programming) For a linear programming problem with constraints that define the polyhedron $\mathcal{P}$ of feasible points we have:
The point $\mathbf{x}_0$is a vertex of $\mathcal{P} \Leftrightarrow$
$\mathbf{x}_0$ is an extreme point of $\mathcal{P} \Leftrightarrow$
$\mathbf{x}_0$ is a BFS (basic feasible solution) of the LP problem.

We now demonstrate part of the proof which will help us in the subsequent discussion about the Simplex Algorithm. We will show that if a point $\mathbf{x}$ is an extreme point of the polyhedron of feasible points $\mathcal{P}$, then $\mathbf{x}$ is BFS of the LP problem.

Assume that $\mathbf{x}$ is extreme point of $\mathcal{P}$. Then it satisfies $\mathbf{x} \in \mathcal{P}$ and $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$. Assume that $\mathbf{x}$ is of the form $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_p, 0, \ldots, 0)^T$, with $p \leq n$. Namely it probably has some of its elements non-zero (if $p = n$ then solution does not have any zero elements). Point $\mathbf{x}$ satisfies the equation

$$x_1\mathbf{a}_1 + \ldots + x_p\mathbf{a}_p = \mathbf{b} \tag{15}$$

where $\mathbf{a}_i$ is the ith column of matrix A. Note that A can be written as $A = [\mathbf{a}_1 \ldots \mathbf{a}_n]$.

Define numbers y_i, for $i = 1, ..., p$ such that:

$$y_1\mathbf{a}_1 + \ldots + y_p\mathbf{a}_p = \mathbf{0}. \tag{16}$$

In order to show that $\mathbf{x}$ is a BFS of the LP problem, it suffices to show that columns $\mathbf{a}_i, i = 1 \ldots p$ of matrix A are linearly independent. If they are, then they will form a basis and the solution $\mathbf{x} = (x_1, \ldots, x_p)$ will be a BFS. Note that the solution is still BFS even if we have the case that $p = n$.

To show $\mathbf{a}_i, i = 1 \ldots p$ of matrix A are linearly independent, we will show that $y_i = 0$ for $i = 1, \ldots, p$. We multiply equation (16) by $\varepsilon > 0$ and add and subtract it from equation (15) . We get:

$$(x_1 + \varepsilon y_1)\mathbf{a}_1 + \ldots + (x_p + \varepsilon y_p)\mathbf{a}_p = \mathbf{b}, \tag{17}$$

and

$$(x_1 - \varepsilon y_1)\mathbf{a}_1 + \ldots + (x_p - \varepsilon y_p)\mathbf{a}_p = \mathbf{b}. \tag{18}$$

Since $x_i > 0$, $\varepsilon > 0$ can be we chosen arbitrarily small arbitrary very small so that $x_i + \varepsilon y_i \geq 0$, and $x_i - \varepsilon y_i \geq 0$. It can be easily deduced that we can choose

$$\varepsilon = \min\left\{\left|\frac{x_i}{y_i}\right|, i = 1, \ldots, p, y_i \neq 0\right\}. \qquad (19)$$

Then, for the vectors

$$\mathbf{z}_1 = (x_1 + \varepsilon y_1, \ldots, x_p + \varepsilon y_p, 0, \ldots, 0) \qquad (20)$$

and

$$\mathbf{z}_2 = (x_1 - \varepsilon y_1, \ldots, x_p - \varepsilon y_p, 0, \ldots, 0) \qquad (21)$$

we have $A\mathbf{z}_1 = \mathbf{b}$, $A\mathbf{z}_2 = \mathbf{b}$, $\mathbf{z}_1, \mathbf{z}_2 \geq \mathbf{0}$, $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{P}$.

Now observe that $\mathbf{x} = \frac{1}{2}\mathbf{z}_1 + \frac{1}{2}\mathbf{z}_2$, but the fact that $\mathbf{x}$ is extreme point results in $\mathbf{x} = \mathbf{z}_1 = \mathbf{z}_2$ which leads to $y_i = 0$ (since $x_i = x_i + \varepsilon y_i \Rightarrow y_i = 0 \ \forall i$). Thus vectors $\mathbf{a}_1, \ldots, \mathbf{a}_p$ are linearly independent and thus $\mathbf{x}$ is a BFS.

As a result of the theorem : if we want to solve an LP problem, we need to search for the optimal solution only among the extreme points of $\mathcal{P}$.

7 Useful facts

Assume we have the system of linear equations $A\mathbf{x} = \mathbf{b}$. Let matrix A have some m linearly independent columns, A is of dimension $m \times n$, $m < n$, and rank$(A) = m$. Denote by B the sub-matrix that consists of these columns. Let D be the submatrix with the rest of the columns. The augmented matrix of this system is $[A\ \mathbf{b}]$. We want to bring this matrix to the form $[I\ D\ \tilde{\mathbf{b}}]$.

We know from the methodology of solving linear systems of equations that the augmented matrix can be brought in that form with elementary operations on it:

- Interchanging any two rows of the matrix,
- Multiplying one of its rows by a real, non-zero number
- Multiplying one of its rows by a real, non-zero number and adding to another row

If the augmented matrix is brought in that form, then $\mathbf{x} = \tilde{\mathbf{b}}$ is the solution of the linear system $A\mathbf{x} = \mathbf{b}$. Because A is of dimension $m \times n$, $m < n$, and rank$(A) = m$ the system has infinite solutions.

Matrix A is brought in the form $[I\ D\ \tilde{\mathbf{b}}]$ and then the linear system of equations can be written as:

$$x_1 + y_{1,m+1}x_{m+1} \cdots y_{1n}x_n = y_{10} \tag{22}$$

$$x_2 + y_{2,m+1}x_{m+1} + \ldots + y_{2n}x_n = y_{20} \tag{23}$$

$\vdots$

$$x_m + y_{m,m+1}x_{m+1} + \ldots + y_{mn}x_n = y_{m0} \tag{24}$$

where the first factor in each equation is reflected in the unit matrix I, the remaining factors in each equation represent matrix D and the right side of the equations represent $\tilde{\mathbf{b}}$.

7.1 Example

We have the following constraints:

$$x_1 + x_2 - x_3 + 4x_4 = 8 \tag{25}$$

$$x_1 - 2x_2 - x_3 + x_4 = 2 \tag{26}$$

The augmented matrix is:

$$\left(\begin{array}{cc} A & \mathbf{b} \end{array} \right) = \left(\begin{array}{cccc|c} 1 & 1 & -1 & 4 & 8 \\ 1 & -2 & -1 & 1 & 2 \end{array} \right)$$

Multiply the first row by -1 and add to the second row:

$$\left(\begin{array}{cc} A & \mathbf{b} \end{array} \right) = \left(\begin{array}{cccc|c} 1 & 1 & -1 & 4 & 8 \\ 0 & -3 & 0 & -3 & -6 \end{array} \right)$$

Divide the second row by -3:

$$\left(\begin{array}{cc} A & \mathbf{b} \end{array} \right) = \left(\begin{array}{cccc|c} 1 & 1 & -1 & 4 & 8 \\ 0 & 1 & 0 & 1 & 2 \end{array} \right)$$

Multiply the second row by -1 and add to the first row.

$$\left(\begin{array}{cc} A & \mathbf{b} \end{array} \right) = \left(\begin{array}{cccc|c} 1 & 0 & -1 & 3 & 6 \\ 0 & 1 & 0 & 1 & 2 \end{array} \right)$$

If we choose as basis matrix $B = [\mathbf{a}_1, \mathbf{a}_2]$, an obvious solution is $\mathbf{x} = (6, 2, 0, 0)$. This solution is feasible (since it satisfies $A\mathbf{x} = \mathbf{b}$), basic and non-degenerate.

If we choose as basis $B = [\mathbf{a}_3, \mathbf{a}_4]$, the solution $\mathbf{x} = (0, 0, 0, 2)$ is feasible, basic and degenerate. Another possible solution is $\mathbf{x} = (3, 1, 0, 1)$ which is feasible but not basic.

If we choose as basis $B = [\mathbf{a}_2, \mathbf{a}_3]$, the solution $\mathbf{x} = (0, 2, -6, 0)$ is basic but non-feasible, since we do not accept negative solutions.

8 Introduction to Simplex Algorithm

In order to solve a Linear Programming problem, we move from one BFS to another BFS until we find the optimal one, which will be the one with the property that if I try to move to whatever other BFs, the value of the objective function is not improved. This is precisely what the Simplex algorithm does.

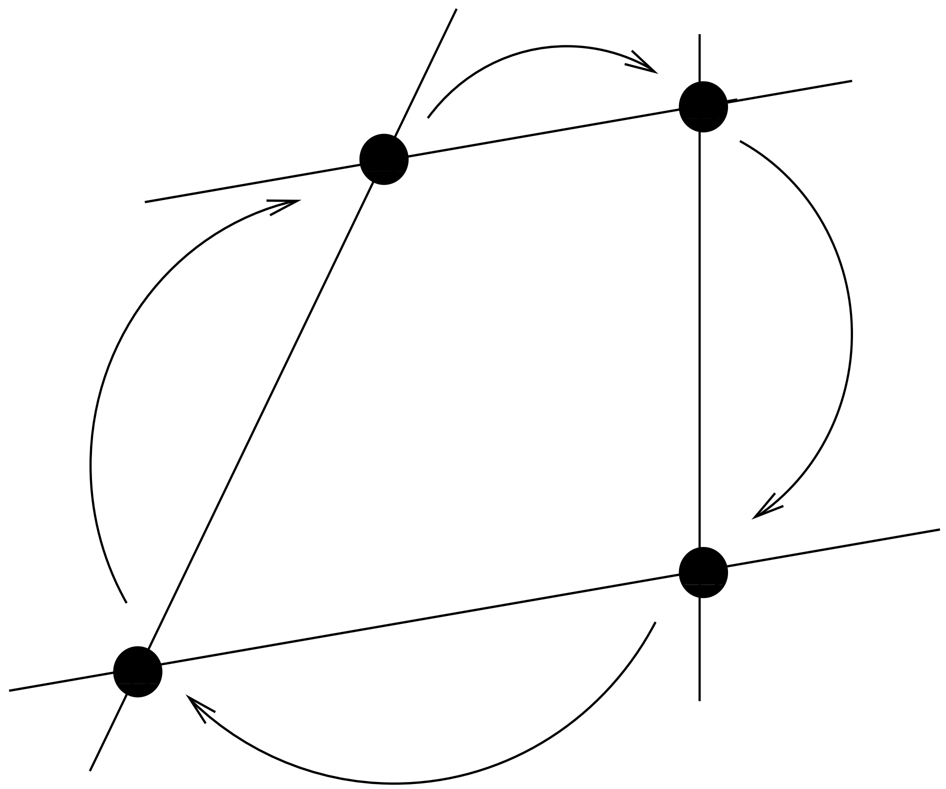

Figure 5: In LP, the Simplex algorithm moves from one BFS to another.

In the following, we will assume that we have non-degenerate solutions. We will treat the cases of degenerate solutions separately.

A given BFS at some step of the algorithm

Consider that at some stage of the algorithm, we have the BFS

$$\mathbf{x} = (y_{10}, \ldots, y_{m0}, 0, \ldots, 0), \;\; y_{i0} > 0, i = 1, \ldots, m. \tag{27}$$

The way, we move from one BFS (vertex, or extreme point of the polyhedron defined by the constraints $A\mathbf{x} = \mathbf{b}$) to another is as follows: In each step we change a non-basic variable to basic and a basic variable to non-basic. This operation is called *pivoting*. The non-basic variables are set to zero while the basic variables are found to be non-negative values. Talking in columns, we insert to the basis a column that is currently not in the basis and we take out of the basis a column that used to be in the basis.

Suppose we have chosen the column $\mathbf{a}_q$, $q > m$ (non-basic column now) and we want to have it inside the base. We have the column $\mathbf{a}_q$ as a linear combination of the current basis $\{\mathbf{a}_1, \ldots, \mathbf{a}_p\}$:

$$\mathbf{a}_q = y_{1q}\mathbf{a}_1 + y_{2q}\mathbf{a}_2 + \ldots + y_{mq}\mathbf{a}_m \tag{28}$$

where matrix $A = [\mathbf{a}_1, \ldots, \mathbf{a}_m, \mathbf{a}_{m+1}, \ldots, \mathbf{a}_q, \ldots, \mathbf{a}_\mathbf{n}]$. Also note that $B = [\mathbf{a}_1 \ldots \mathbf{a}_m]$ and $D = [\mathbf{a}_{m+1}, \ldots, \mathbf{a}_n]$

We want to move the non-basic column $\mathbf{a}_q$ in the basis. Multiply the left- and the right-hand side of the equation above with $\varepsilon > 0$ to get:

$$\varepsilon\mathbf{a}_q = \varepsilon(y_{1q}\mathbf{a}_1 + \ldots + y_{mq}\mathbf{a}_m) \tag{29}$$

Now we know that the current BFS (the vector $\tilde{\mathbf{b}}$ we saw before) satisfies $A\mathbf{x} = \mathbf{b}$ and can be written as a linear combination of the basic columns as:

$$y_{10}\mathbf{a}_1 + \ldots + y_{m0}\mathbf{a}_m = \mathbf{b} \tag{30}$$

Subtract the one equation from the other to get:

$$\mathbf{b} = \\ (y_{10} - \varepsilon y_{1q})\mathbf{a}_1 + (y_{20} - \varepsilon y_{2q})\mathbf{a}_2 + \dots \\ +(y_{m0} - \varepsilon y_{mq})\mathbf{a}_m + \varepsilon \mathbf{a_q} \quad (31)$$

Notice that since the equation $A\mathbf{x} = \mathbf{b}$ is satisfied again, the coefficients correspond to a new solution,

$$\begin{pmatrix} y_{10} - \varepsilon y_{1q} \\ y_{20} - \varepsilon y_{2q} \\ \vdots \\ y_{m0} - \varepsilon y_{mq} \\ 0 \\ \varepsilon \\ 0 \end{pmatrix}$$

where ε appears in the q-th position, $q > m$.

This operation can be understood as follows: currently $x_q = 0$. Assume we start increasing the (currently non-basic) variable x_q to some positive value ε, so as to make it basic. Equivalently, column $\mathbf{a}_q$ will enter the basis. When ε increases, then q-th component increases too. The variables $x_1, \dots, x_m$ decrease if $y_{iq} > 0$ and increase if $y_{iq} < 0$.

The question that arises now is: Up to which value ε can x_q be increased so that we go from one BFS to another? The answer is the following: the maximum value that ε can take is determined by that component of the solution that becomes zero first. This value of ε is clearly

$$\varepsilon = \min_i \left\{ \frac{y_{i0}}{y_{iq}} : y_{iq} > 0 \right\} \tag{32}$$

Suppose that p is the index that is first zeroed, $1 \leq p \leq m$. Then, p corresponds to the basic variable that will now become non-basic. Specifically, it is

$$p = \arg \min_{i=1,\ldots,m} \left\{ \frac{y_{i0}}{y_{iq}} : y_{iq} > 0 \right\} \tag{33}$$

Therefore, variable x_p now becomes zero, or equivalently column $\mathbf{a}_p$ exits the basis. The new basis is therefore,

$$\{\mathbf{a}_1, \ldots, \mathbf{a}_{p-1}, \mathbf{a}_{p+1}, \ldots, \mathbf{a}_m, \mathbf{a}_q\} \tag{34}$$

and the new BFS is:

$$\begin{pmatrix} y_{10} - \varepsilon y_{1q} \\ \vdots \\ y_{p-1,0} - \varepsilon y_{p-1,q} \\ 0 \\ y_{p+1,0} - \varepsilon y_{p+1,q} \\ \vdots \\ y_{m0} - \varepsilon y_{mq} \\ 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{pmatrix}$$

with ε in the q-th position.

8.1 Cases of degenerate BFS

It may happen that the new BFS is degenerate, namely a variable that is basic is zero. This can happen in the following cases:

1. The coefficients are such that two basic variables can become zero when we want to change the base. For the

example above, we may have two indices $0 \leq p_1, p_2 \leq m$ such that $x_{p_1} = 0$ and $x_{p_2} = 0$.

2. It may happen that $\varepsilon = 0$. Then, the variable we want to turn to basic and place it in the basis cannot take a larger value and the BFS remains the same. When the current BFS $\mathbf{x}$ is degenerate, then it may be that $\varepsilon = 0$ and the new BFS remains the same as the current BFS (especially this is the case if some basic variable is 0 and the corresponding denominator is positive).

3. Cycling phenomenon : It may happen that the a sequence of basis changes lead us through a sequence of changes of bases back to the initial basis.

Also note that in the case that none of coefficients y_{iq} is positive, we can move further from the current BFS, but we cannot discover any new BFS for the problem $\Rightarrow$ the polyhedron *P* is unbounded and the LP problem is said to be unbounded.

8.2 Critical questions

8.2.1 When does Simplex algorithm stop?

Suppose that we have a BFS

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_B \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} y_{10} \\ y_{20} \\ \vdots \\ y_{m0} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

What is the cost of the solution? In other words, we find the value of the objective function $\mathbf{c}^T\mathbf{x}$ for this solution. We have the value

$$z = \mathbf{c}^T\mathbf{x} = c_1 y_{10} + c_2 y_{20} + \ldots + c_m y_{m0} = \mathbf{c}_B^T\mathbf{x}, \quad (35)$$

where $\mathbf{c}_B^T$ is the part of the cost vector $\mathbf{c}$ that corresponds to the basis. Suppose we get to a new BFS,

$$\mathbf{x}^{'} = \begin{pmatrix} y_{10} - \varepsilon y_{1q} \\ y_{20} - \varepsilon y_{2q} \\ \vdots \\ y_{m0} - \varepsilon y_{mq} \\ 0 \\ \vdots \\ \varepsilon \\ \vdots \\ 0 \end{pmatrix}$$

where ε comes at the q-th component of the solution. The new cost is:

$$z^{'} = \sum_{i=1, i \neq p}^{m} c_i (y_{i0} - \varepsilon y_{iq}) + c_q \varepsilon, \tag{36}$$

where q denotes the new variable x_q that became a basic variable. We have

$$z^{'} = z + \varepsilon [c_q - (c_1 y_{1q} + c_2 y_{2q} + \ldots + c_m y_{mq})]. \tag{37}$$

Now we set z_q the new column that we want to insert to the basis as a function of the old basis:

$$z_q = c_1 y_{1q} + c_2 y_{2q} + \ldots + c_m y_{mq}. \tag{38}$$

Then we get

$$z' = z + \varepsilon(c_q - z_q) = z + \varepsilon\ r_q \tag{39}$$

or $z' - z = \varepsilon(c_q - z_q)$.

In order for the new solution to be ”better” than the current one, its objective function value has to be smaller than the one of the current solution. If $z' - z < 0$, then the new BFS (that corresponds to the non-basic column $\mathbf{a}_q$ entering the basis) has a lower objective function value. Since $\varepsilon > 0$, this happens when $c_q - z_q < 0$ is true. Define $c_q - z_q = r_q$ as the *reduced cost coefficient* corresponding to the newly entered variable x_q. If $c_q - z_q < 0$ then by entering column $\mathbf{a}_q$ in the basis, we have arrived to a better BFS (one with a lower cost).

Fact: The solution

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_B \\ 0 \end{pmatrix}$$

is *optimal* if $\forall q = m+1, \ldots, n$ it is $c_q - z_q > 0$. In other words, this means that I am currently at the optimal BFS if I try to change the basis by all possible means (i.e, insert to the basis *any* non-basic column $\mathbf{a}_q$) and I will never manage to reduce the value of the objective function. Note that the reduced cost variables $c_q - z_q = r_q$ are defined for each variable. As an exercise, we can easily show the following:

Problem: Show that for the basic variables $x_i,\ i = 1, \ldots, m$, it is $r_i = 0$.

Remark: The change of basis performed between columns $\mathbf{a}_p$ and $\mathbf{a}_q$ is called (p,q) *pivoting operation*.

8.2.2 How do I choose which non-basic variable will become basic?

How can I select the column q that will enter the base? There are three possible ways to do that:

1. If $z' - z = \epsilon(c_q - z_q)$, we choose q such that the difference $c_q - z_q$ takes the minimum value (the rate of cost reduction becomes as large as possible). To be more specific,

$$q = \arg \max_{l \in \{m+1,\ldots,n\}} |r_l| = \arg \min_{l \in \{m+1,\ldots,n\}} r_l. \tag{40}$$

   Thus, we choose to make basic the variable that leads to the largest rate of cost reduction (cost reduction per unit of non-basic variable increase, $c_q - z_q = \frac{z'-z}{\varepsilon}$).

2. Choose

$$q = \arg \max_{l \in \{m+1,\ldots,n\}} \varepsilon_l |r_l| = \arg \min_{l \in \{m+1,\ldots,n\}} \varepsilon_l r_l. \tag{41}$$

   In this way of choosing q we take into account the *actual* change in the cost value, which also depends on ε. Note that ε_l is the value of ε that turns the non-basic variable x_l to basic and thus clearly ε has a different value for different q. Indeed with this way of selecting q, we reduce the value of the objective function as much as possible, since $\varepsilon_l(c_l - z_l) = z' - z$. Of course there is the additional computational burden of finding ε_l and computing the products $\varepsilon_l r_l$.

3. Choose $q = \arg\min_{i \in \{m+1,\ldots,n\}}\{r_i : r_i < 0\}$. That is, we examine all non-basic variables starting from the lowest-indexed one and select to place at the basis the first one that has negative reduced cost coefficient. The disadvantage with this approach is that it does not guarantee that the value of the objective function at the new BFS will be the smallest possible.

If we choose q using way (3), and choose $p = \min\{j : \frac{y_{jo}}{y_{jq}}\} = \min_i\{\frac{y_{io}}{y_{iq}}\} : y_{iq} > 0\}$ (in other words, we choose to put out of the basis the lowest-indexed variable out of the ones that become 0), then, even though I have a non-degenerate new BFS, the cycling phenomenon mentioned above is avoided. This is known as *lexicographic pivoting rule*.

8.3 Simplex algorithm : steps

1. Begin with an initial BFS. If the LP problem is defined using inequalities, we define slack variables and bring it to the standard form and find the BFS.

2. Calculate the coefficients r_q for each non-basic variable x_q.

3. - If $r_q \geq 0$ for all non-basic variables, then STOP the algorithm. We have found the optimal solution.
   - Else, choose q according to one of the rules that we described in question (2) previously.
     If none of $y_{iq} > 0$ then STOP(unbounded LP problem)
     Else, calculate $p = \arg\min_i\{\frac{y_{i0}}{y_{iq}} : y_{iq} > 0\}$ (this selection rule for the variable that exits the basis eliminates the cycling phenomenon).

4. Pivot(p,q) and find new BFS.

5. Go to step 2.

8.4 Example of Simplex Algorithm for an LP problem

Given the LP problem

$$\max 7x_1 + 6x_2$$

subject to the constraints:

$$2x_1 + x_2 \leq 3 \quad (42)$$
$$x_1 + 4x_2 \leq 4 \quad (43)$$
$$x_1, x_2 \geq 0. \quad (44)$$

solve it (find the optimal BFS).

Solution: We convert the problem to the standard form by defining slack variables x_3, x_4, so the new problem (P) is:

$$\min -7x_1 - 6x_2$$

subject to:

$$2x_1 + x_2 + x_3 = 3 \quad (45)$$
$$x_1 + 4x_2 + x_4 = 4 \quad (46)$$
$$x_1, x_2, x_3, x_4 \geq 0 \quad (47)$$

The objective function value z is : $z = -7x_1 - 6x_2 + 0x_3 + 0x_4$.

We start running the Simplex Algorithm. Start with initial BFS: $\mathbf{x} = (0, 0, 3, 4)$. The basis is $B = [\mathbf{a}_3, \mathbf{a}_4]$.

At each step, we will express the cost as a function of the non-basic variables. We will also write the basic variables as functions of the non-basic variables to facilitate computation of ε and the pivoting.

Initial cost:$z = -7x_1 - 6x_2 + 0x_3 + 0x_4$ with value $z_0 = 0$ for the current BFS. Write the basic variables as functions of the non-basic ones:

$$x_3 = 3 - 2x_1 - x_2 \tag{48}$$

$$x_4 = 4 - x_1 - 4x_2 \tag{49}$$

STEP 1: Our goal is to change the basis, so that we find a new BFS with lowest cost value. The question is ”which (non-basic) variable x_p should we choose to make basic?”. We choose it according to the first rule case out of the three we described at question (2) in this lecture.

We see that $r_1 = -7$, $r_2 = -6$ (reduced cost coefficients). In other words, if I increase x_1 or x_2 by making one of the two basic, I observe that the increase of x_1 causes the largest decrease in z. That is, since $|r_1| > |r_2|$, we choose to make variable x_1 basic (equivalently put the first column $\mathbf{a}_1$ of matrix A in the basis. Thus it is $q = 1$. If we increase x_1, we observe that, out of the basic variables x_3, x_4, the first that becomes zero is x_3 and this occurs for $x_1 = \varepsilon = 3/2$. Thus $p = 3$ and x_3 will become non-basic.

Pivot$(3, 1)$. New basis: $B = [\mathbf{a}_1, \mathbf{a}_4]$

New BFS: $\mathbf{x} = (\frac{3}{2}, 0, 0, \frac{5}{2})$ and new cost value: $z = -\frac{21}{2}$ (observe that we reduced the value of the objective).

We now write the new basic variables as functions of non-basic variables:

$$x_1 = \frac{3}{2} - \frac{1}{2}x_2 - \frac{1}{2}x_3 \tag{50}$$

$$x_4 = \frac{5}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 \tag{51}$$

and the cost:

$$z = -7x_1 - 6x_4 = -\frac{21}{2} - \frac{5}{2}x_2 + \frac{7}{2}x_3 \tag{52}$$

STEP 2: Now again we will have to choose which non-basic variable to make basic. Observe that if we make variable x_3 basic, the cost will be increased, which is undesirable. So we choose to make basic the variable x_2. Thus $q = 2$.

If we increase x_2, we observe that, out of the basic variables x_1, x_4, the first that becomes zero is x_4 and this occurs for $x_2 = \varepsilon = 5/7$. Thus $p = 4$ and x_4 will become non-basic.

Pivot$(4, 2)$. New basis: $B = [\mathbf{a}_1, \mathbf{a}_2]$

New BFS: $\mathbf{x} = (\frac{8}{7}, \frac{5}{7}, 0, 0)$ and new cost value: $z_2 = -\frac{81}{7}$ (observe that with the change of basis we have reduced the objective function value more). From equations

Again, we write the basic variables as a function of non-basic variables.

$$x_1 = \frac{8}{7} - \frac{4}{7}x_3 + \frac{1}{7}x_4 \tag{53}$$

$$x_2 = \frac{5}{7} + \frac{1}{7}x_3 - \frac{2}{7}x_4 \tag{54}$$

and the cost:

$$z = -7x_1 - 6x_2 = -\frac{86}{7} + \frac{22}{7}x_3 + \frac{5}{7}x_4 \tag{55}$$

STEP 3: Now again we will have to choose which non-basic variables to make basic. However, observe that if make either x_3 or x_4 basic (i.e try to increase them from 0), the cost value will be increased. So the algorithm stops here and we say that we found the optimal BFS and we solved the LP problem.

Optimal solution: $\mathbf{x} = (\frac{8}{7}, \frac{5}{7}, 0, 0)$.

An alternative presentation of the above steps: We can also find the solution using the matrix form. In every step, the p and q variables are chosen the same way as we already mentioned above. Thus, we have:

The augmented matrix of the primal problem is:

$$\begin{pmatrix} A & \mathbf{b} \end{pmatrix} = \left(\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 3 \\ 1 & 4 & 0 & 1 & 4 \end{array}\right)$$

So, the basis is $B = [\mathbf{a}_3, \mathbf{a}_4]$ and thus an initial BFS is again $\mathbf{x} = (0, 0, 3, 4)$.

STEP 1: Following the same procedure as above we find that the new basis must be $B = [\mathbf{a}_1, \mathbf{a}_4]$ in order to lower cost

value.
Multyplying the first row with $-\frac{1}{2}$ and it add to the second row and then divide the first row by 2, we take the new basis:

$$\begin{pmatrix} A & \mathbf{b} \end{pmatrix} = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} & 0 & | & \frac{3}{2} \\ 0 & \frac{7}{2} & -\frac{1}{2} & 1 & | & \frac{5}{2} \end{pmatrix}$$

So, the new BFS in step1 is again $\mathbf{x} = (\frac{3}{2}, 0, 0, \frac{5}{2})$.

STEP 2: In the same spirit like before, we find that the new basis must be $B = [\mathbf{a}_1, \mathbf{a}_2]$ so as to have additional reduction of the cost value. Multyplying the first row with $-\frac{1}{7}$ and it add to the second row and then divide the first row by $\frac{2}{7}$, we take this new basis:

$$\begin{pmatrix} A & \mathbf{b} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \frac{4}{7} & -\frac{1}{7} & | & \frac{8}{7} \\ 0 & 1 & -\frac{1}{7} & \frac{2}{7} & | & \frac{5}{2} \end{pmatrix}$$

So, the new BFS in step2 is again $\mathbf{x} = (\frac{8}{7}, \frac{5}{7}, 0, 0)$.

STEP 3: Like before, the algorithm stops here and we say that we found the optimal BFS: $\mathbf{x} = (\frac{8}{7}, \frac{5}{7}, 0, 0)$., because we again observe that the cost value will be increased if make either x_3 or x_4 basic.

9 Modelling Network Problems using LP

9.1 Maximum lifetime routing in wireless sensor networks

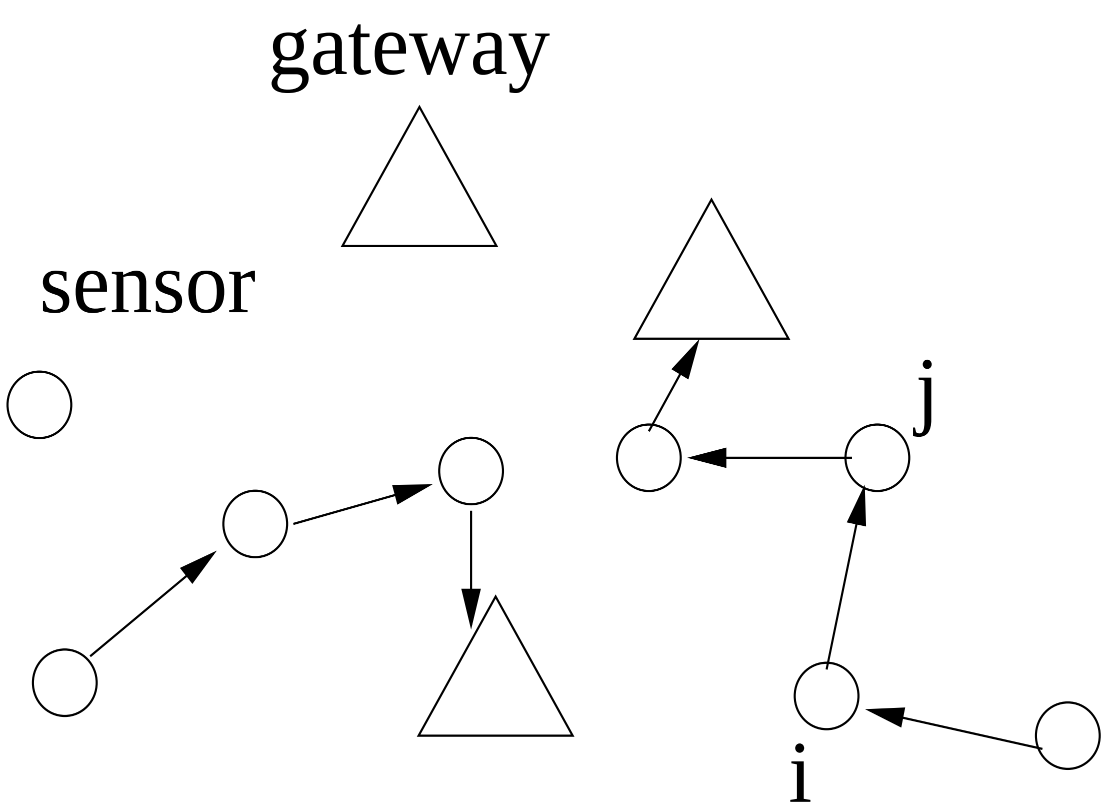

Figure 6: A wireless sensor network with a set of sensor nodes and a set of gateways.

We will now see an example of a problem from wireless sensor networks, that is formulated as an LP problem.

A sensor network consists of a set of miniature-sized sensor nodes that sense and monitor processes such as vibration, sound, light, temperature, movement. The information needs to be transferred from the sensors to a set of information processing centers, called gateways. The data will be transferred with multi-hop routing as shown in figure 1. Of course the different routes may intersect (something that is not shown in the figure). Sensor nodes have the ability to forward their data, as well as other nodes' data.

All sensors can potentially produce information data. Gateways are connected with optical fiber and we will assume that the data from a sensor will have reached its destination if it reaches *any* of the gateways. We will also not deal with interference from multiple ongoing transmissions (which can be here assume to be reduced or eliminated by means of an appropriate scheduling protocol). We assume there exist several *kinds* (or *commodities*) of traffic (e.g. temperature, sound, etc)

The transmitted energy from a sensor node can be adjusted to a level appropriate for a receiver within its transmission range to the able to receive the data correctly. We will discuss later this issue.

Upon arrival of new information at a node (either generated by the node itself or forwarded from other nodes) a routing decision needs to be made so that the data is forwarded to an appropriate neighbor. We will see that routing accounts to finding the way to split the traffic streams across different routes, so as to ”balance” energy consumption among nodes and thus increase network lifetime.

The topology is considered to be static, namely with no mobility. Note that mobility either of the sensor nodes or of the gateways or both may further help in reducing the energy consumption and improve network lifetime. However, we will not consider such an issue here.

We define the following quantities:

P_{ij} : the minimum power needed for sensor i to send information to to sensor j. This os proportional to the distance d_{ij} between the nodes i and j and is given as $P_{ij} = \gamma d_{ij}^a$, where a is a constant that specifies the type of wireless propagation environment and γ is the minimum required SNR at the receiver such that reception is acceptable. sensors

Specifically, at the receiver we have SNR $\geq \gamma \Rightarrow \dfrac{P_{ij}}{d_{ij}^a \sigma^2} \geq \gamma$, and thus the minimum power is $P_{ij} = \gamma d_{ij}^a \sigma^2$. Note that σ^2 is the noise power.

e_{ij} : The amount of energy consumed by sensor i for the transport of one unit of information (bit or packet) to sensor j. It is measured in Joules per bit. Now, we determine e_{ij} in more detail and note that e_{ij} is known to each sensor i only for its neighbors j (let $\mathcal{N}_i$ be the set of neighbors of sensor node i).

Energy and power are related as $E = Pt$, which has units (energy/bit) = (power/bit) $\times$ (sec). We understand that energy per bit is the product of power and transmitting rate, $e_{ij} = P_{ij} r_{ij}$. Thus, the parameter e_{ij} captures *both* changing the power level and the transmission rate, e.g changing the modulation level to reach j.

Each sensor has an initial amount of energy E_i. During a sensor operation, energy can be consumed to perform the following tasks:

- Transmission of information (this is the most energy-consuming task).
- Reception of information (since the reception circuits have to be on and process the received information).
- CPU operation (battery is consumed to perform numerical operations and tasks. Thus, the algorithms for sensor networks need to be simple and of low computational load.
- Sensing (the sensing module consumes energy)
- ON-time of circuits. Even if not involved in any of the operations above, a sensor consumes energy even by being ON (awake as we say).

In this problem, we will assume that energy is consumed only for transporting data. The network can be represented as a directed graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of edges. An edge exists whenever a node j is within the transmission range of node i (can be reached for a constant power P)

We now define the following quantities that will help us construct the model in our problem. Let $c = 1, \ldots, C$ denote the C different kinds of information transferred in the network.

- $Q_i^{(c)}$: Rate of generation of kind c of traffic at node i in units bits/sec.
- $Q_i = \sum_{c=1}^{C} Q_i^{(c)}$: total rate traffic generation at node i.
- $q_{ij}^{(c)}$: The rate at which information of kind c is transferred from node i to j (in bits/sec).
- $q_{ij} = \sum_{c=1}^{C} q_{ij}^{(c)}$: The total rate of information transfer from sensor i to sensor j (in bits/sec).
- $O_c = \{i \in \mathcal{N} : Q_i^{(c)} > 0\}$. The set of origin nodes of traffic of type c.
- D_c : the set of destinations of traffic of type c

The variables are the q_{ij}'s. The problem of routing is equivalent to finding flows q_{ij} or equivalently the flow vector $\mathbf{q} = \{q_{ij}\ \forall (i,j) \in \mathcal{A}\}$. Vector $\mathbf{q}$ shows how information flows in each edge (i,j) and the way that information streams are split in each node.

9.1.1 Network Lifetime

The sensor network lifetime is defined as time between the beginning of network operation and the time when first node ”dies”, namely its energy vanishes and its battery is drained. Here, we should note that there are several different definitions of network lifetime. For example, we lifetime can be alternatively defined as the time until transfer of information from the sources to the destinations is still feasible no matter how many nodes have zero battery. Or network lifetime, can be the time when the battery of some percentage $k\%$ of sensor nodes becomes zero. However, we will consider the definition of network lifetime we said before. This is a meaningful definition in the following sense: the network operates normally until the first node’s battery finishes. Then, this node cannot handle traffic any more and additional re-routing algorithms need to be applied in the network to circumvent that node and find alternative routes. Hence, much more additional energy is needed and the rate at which nodes’ batteries will be emptying will be higher from then on. Therefore, the time when the battery of one node vanishes is a benchmark and can be defined as the network lifetime.

Let us express the network lifetime as a function of flow vector $\mathbf{q}$. First, we express the *node lifetime* as a function of the flow vector. In the problem formulation from now on, we will assume that the network carries one type of traffic. The generalization to more than one types of traffic is easy.

For a flow vector $\mathbf{q}$, the node lifetime is:

$$T_i(\mathbf{q}) = \frac{E_i}{\sum_{j \in \mathcal{N}_i} e_{ij} q_{ij}} \tag{56}$$

where the denominator shows the total rate of decrease of energy for node i.

The *Network lifetime* for flow vector $\mathbf{q}$ is defined as:

$$T_N(\mathbf{q}) = \min_{i \in \mathcal{N}} T_i(\mathbf{q}) = \min_{i \in \mathcal{N}} \frac{E_i}{\sum_{j \in \mathcal{N}_i} e_{ij} q_{ij}} \tag{57}$$

The maximum lifetime routing problem is defined as follows: Find the flow vector $\mathbf{q}$ so as to maximize network lifetime:

$$\max_{\mathbf{q}} T_N(\mathbf{q}) = \max_{\mathbf{q}} \min_{i \in \mathcal{N}} \frac{E_i}{\displaystyle\sum_{j \in \mathcal{N}_i} e_{ij} q_{ij}} \tag{58}$$

for $q_{ij} \geq 0, \forall\, i \in \mathcal{N},\ \forall j \in \mathcal{N}_i$.

There is also the following constraint for the problem:

$$Q_i + \sum_{j : i \in \mathcal{N}_j} q_{ij} = \sum_{k \in \mathcal{N}_i} q_{ik},\ \forall\, i \in \mathcal{N} \setminus D^c \tag{59}$$

This equation expresses the flow conservation principle at each node i. The way the problem is formulated now is not Linear and so is difficult to solve. With a change of variables and the definition of a new variable, we will show that the formulation above is actually equivalently to a Linear Programming problem. Define the network lifetime as a new variable,

$$T = \min_{i \in \mathcal{N}} \frac{E_i}{\sum_{j \in \mathcal{N}_i} e_{ij} q_{ij}} \tag{60}$$

Then, since T is the mimimum of all lifetimes for every i, we have:

$$T \leq \frac{E_i}{\sum_{j \in \mathcal{N}_i} e_{ij} q_{ij}}, \ \forall i \tag{61}$$

Multiplying the flow conservation equation with T we get:

$$TQ_i + T \sum_{j: i \in \mathcal{N}_j} q_{ij} = T \sum_{k \in \mathcal{N}_i} q_{ik} \tag{62}$$

We define new variables $\hat{q}_{ij} = Tq_{ij}$ where $\hat{q}_{ij} \geq 0$. Then, the flow conservation equation becomes:

$$TQ_i + \sum_{j: i \in \mathcal{N}_j} \hat{q}_{ij} = \sum_{k \in \mathcal{N}_i} \hat{q}_{ik}. \tag{63}$$

Also, there appears the inequality constraint:

$$\sum_{j \in \mathcal{N}_i} e_{ij} \hat{q}_{ij} \leq E_i. \tag{64}$$

The objective function is now linear in the variable vector $(\hat{\mathbf{q}}, T)$ and the objective is now stated as

$$\max_{\hat{\mathbf{q}},T} T \tag{65}$$

Thus, we converted our problem into a linear programming problem, since the objective is linear in the variable vector and the constraints are linear in the variables as well.

In general if we have a problem of the form min-max (the problem above was of the max-min form)

$$\min_{\mathbf{x}} \max_{i=1,\ldots,m} \mathbf{a}_i^T \mathbf{x} + b_i, \tag{66}$$

the idea is to define an extra variable,

$$t = \max_{i=1,\ldots,m} \mathbf{a}_i^T \mathbf{x} + b_i. \tag{67}$$

and the problem will be:

$$\min_{\mathbf{x},t} t \tag{68}$$

subject to the constraints:

$$t \geq \mathbf{a}_i^T \mathbf{x} + b_i, \forall\, i = 1, \ldots, m \tag{69}$$

When we have several linear functions of $\mathbf{x}$, the min-max problem is converted into a linear programming problem.

9.2 Carrier assignment in OFDM systems

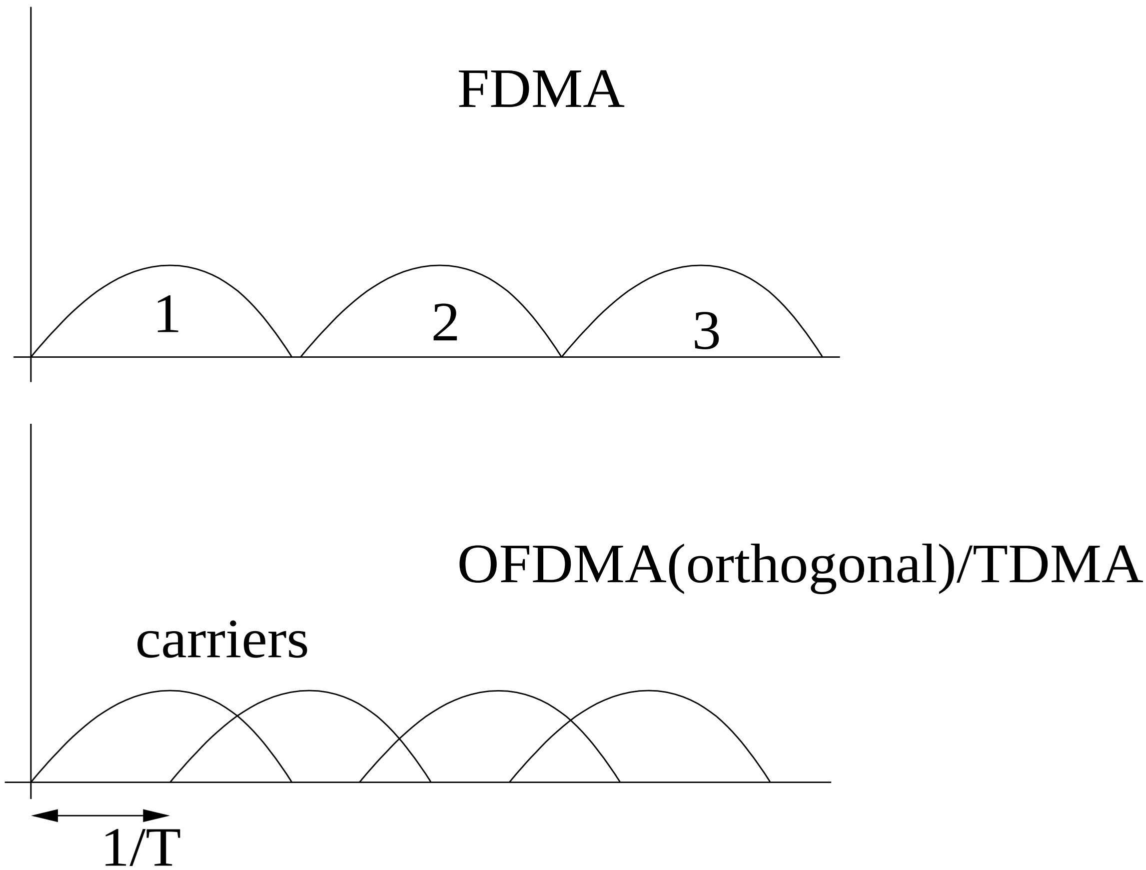

Figure 7: OFDM versus conventional FDMA.

We will now examine and formulate the problem of carrier assignment in OFDM systems as a linear programming problem. OFDM (Orthogonal Frequency Division Multiplexing) is different from conventional FDMA systems in the following sense: in OFDM systems, the spectrum is divided into several subcarriers with overlapping spectra (see figure 7 . Note than in FDMA, the spectrum is divided into non-overlapping spectra.

The innovation in OFDM is that, due to a well-known property of the Fourier transform, although the spectra are overlapping they are *orthogonal* to each other, that is, they do not cause interference to each other. Thus, more efficient use of spectrum is achieved. The OFDM is said to have higher *spectral efficiency* than FDMA. Another innovation is that each user

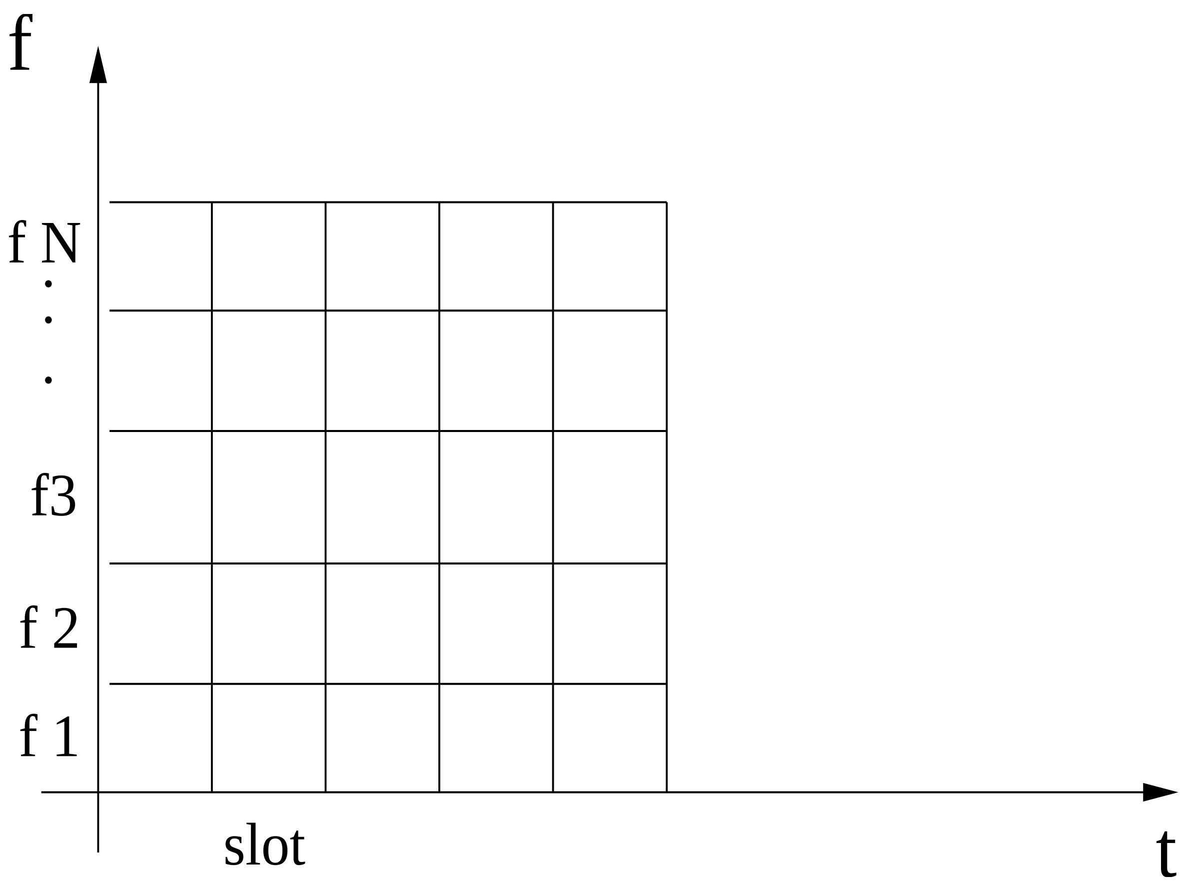

Figure 8: Resources in the system : Timeslots making up one frame and sub-carriers

can split its bit stream and use several sub-carriers in parallel (each sub-carrier corresponds e.d. to a different frequency. Note that in FDMA, each user was allocated one frequency.

We will consider here an OFDM/TDMA system. There exist

K users and N subcarriers. Time is divided into time slots and C time slots make up one time frame. There are two kinds of resources to be allocated to users: the frequencies (subcarriers $1, \ldots, N$) and the timeslots (slots $1, \ldots, C$ within a frame). See figure 8 for how resources are organized.

9.2.1 Formulation of a sub-carrier assignment problem as an LP problem

This example is about carrier assignment to users in one cell. the BS transmits to K users with N subcarriers in the down-link.

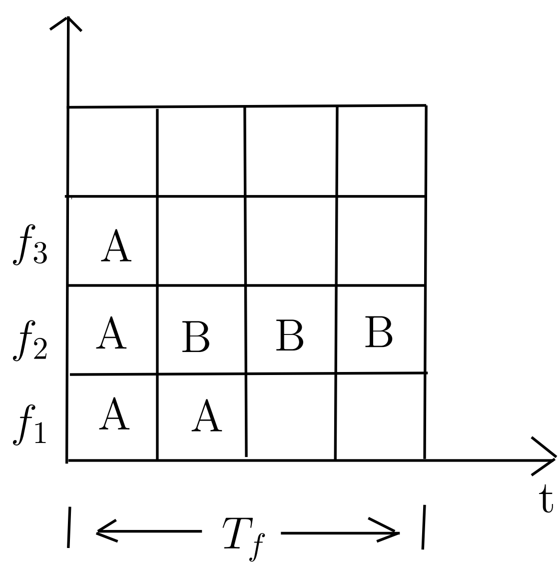

Figure1 : Example
T_f : frame duration

There exist a set of subcarrier frequencies and a set of C time slots that need to be assigned to users in an OFDM system. The timeslots make up a time frame of duration T_f sec. In general, a user can be assigned several time slots in several different subcarriers. The figure below shows the resources (subcarriers/timeslots) that are assigned to two users A and B.

Each user i perceives each subcarrier j to be of different quality. There exist two main reasons for that:

- Co-channel interference. Different users are located in different regions within a cell and thus experience different amounts of co-channel interference in each sub-carrier due to different amounts of subcarrier reuse in neighboring cells. For example, a user may be in a location that is close to a cell in which subcarriers 1 and 2 are reused which subcarrier 3 is not. In that case, that user perceives subcarrier 3 as being of very good quality, while subcarriers 1 and 2 are of lower quality. Also, different users perceive the *same* subcarrier as being of different quality for the same reason. For example, if another user is in a location close to a cell where subcarrier 3 is used, then this user perceives subcarrier 3 as of lower quality than the first user does.

- Frequency selectivity. Even in the absence of interference whatsoever, a user has different channel gain in different frequencies. That is, a transmitted signal has different channel attenuation in different frequencies. Frequency selectivity results in different *frequency response function* $H(f)$ in different subcarrier frequencies f. The phenomenon of frequency selectivity is attributed to multipath: the same signal follows several different paths while traveling from the transmitter to the receiver and each path is of different length (and thus arrives with different delay at the receiver). These delay differences give rise to a frequency dependence on the amplitude and phase of the signal (for more details, look in the class of Wireless Communications notes).

In the problem, we assume that the time frame duration is small enough, so that each user has the same quality across all time slots of a subcarrier. Therefore, we do not consider temporal channel quality variations in our problem.

Given the fact that different users experience different quality in a subcarrier, there comes the question: *Which user is the most suitable to be assigned to a subcarrier*?

As mentioned above, each user i experiences different quality in different subcarriers j. If the user utilizes only one carrier j, the achieved transmission rate r_i is:

$$\begin{aligned} r_i &= \frac{S}{T_f} \cdot b_{ij} \cdot a_{ij} \cdot C \cdot T_f \\ &= S \cdot b_{ij} \cdot a_{ij} \cdot C \end{aligned} \tag{70}$$

where S is the number of symbols of the user transmitted in a time slot, a_{ij} is the number of time slots that are used in a frame in subcarrier j , b_{ij} is the maximum tolerable modulation level (in bits/symbol) that can be assigned to a user in subcarrier j and $C \cdot T_f$ is the time that is needed per frame (sec/frame). Rate r_i has units of bits/sec. Clearly, the achievable rate for a user i in a subcarrier j depend on b_{ij}, a_{ij}. The more slots the user uses, the more bits it can transfer in a frame duration. Also, the larger the modulation level, the larger the achievable rate. The quality of each subcarrier j for a user i is reflected on the Signal-to-Interference and Noise ratio (SINR) of a user.

As we have said before, if the SINR is large (the subcarrier is of good quality), the BS can afford to transmit with a large modulation level (and still maintain the BER below a threshold ϵ. On the other hand, if the SINR for a user in a subcarrier is not good, the BS cannot transmit with a high modulation level. Instead, it needs to use lower modulation level, so as to maintain BER below ϵ.

Each user sends a message to the BS and informs it about the quality in each subcarrier. The user receiver can easily measure the quality in different subcarriers at its receives and then it can feed back this information at the BS. The user i thus essentially indirectly informs the BS of the modulation level vector $(b_{i1}, ..., b_{iN})$ that the BS can give in the different subcarriers. Now, if the user has also declared its requirements in rate to the BS, the BS gets informed about the user's preferences and can estimate how many slots are needed for every user in order to fulfil its rate requirements (if the user uses exclusively one subcarrier). Thus, the number of slots that are needed by a user i in order to fulfil its rata requirements if it used only one subcarrier j is:

$$a_{ij} = \left\lceil \frac{r_i \cdot T_f}{S \cdot b_{ij}} \right\rceil \quad (71)$$

Thus we make the following observations:

- The more the user rate requirements, the more the slots that the user needs in order to fulfil them.

- The better the channel quality in a subcarrier for a user, the higher the achievable modulation level and thus the fewer the number of slots that are required in order for the user to fulfil its rate requirements.

The BS faces the following problem:
Given a number of users with some rate requirements and given a number of carriers, allocate carriers and timeslots to users, such that the user rate requirements are satisfied and the minimum total number of time slots are used. This optimization objective is meaningful, since the BS would like to have as many free slots are possible in case they are needed:

- in order to serve a burst of many new arriving users. Hence the free slots help to serve sudden increased in user loads.

- in order to cope with sudden subcarrier quality deterioration for many users. This often occurs in wireless systems. If the quality of one or more subcarriers deteriorates for users, then the users need additional time slots in order to fulfil rate requirements.

An example with three users A,B and C and three subcarrier frequencies f_1, f_2, f_3 is given in the figure below (denoted as Figure 4). The matrix element (i, j) shows the number of required time slots by user i in subcarrier j in order to fulfil its rate requirements by using *exclusively* this subcarrier. Thus for example user A prefers to use subcarrier f_1 since it will occupy fewer slots there. User C also prefers subcarrier f_1 to the other two for the same reason.

	f_1	f_2	f_3
A	2	5	3
B	4	1	2
C	3	4	5

Figure 4 : User requirements in frequencies

However, if a subcarrier is preferable by several users, it may happen that the number of slots in that subcarrier is not adequate to accommodate all users. For instance, if subcarrier f_1 has $C = 4$ time slots in a frame, and user A is assigned to subcarrier f_1, user B to subcarrier f_2 and user C to subcarrier f_1, then 5 slots are needed to satisfy users A and C in subcarrier f_1 (but only 4 are available!). This is shown in the figure below (denoted as figure 5)

So,only 2 slots out of the 3 needed can be given to user C at subcarrier f_1. Thus the rest of its requirements need to be fulfilled by assigning to the user slots by lesser quality subcarrier (e.g the next more preferable subcarrier for user C is f_2).

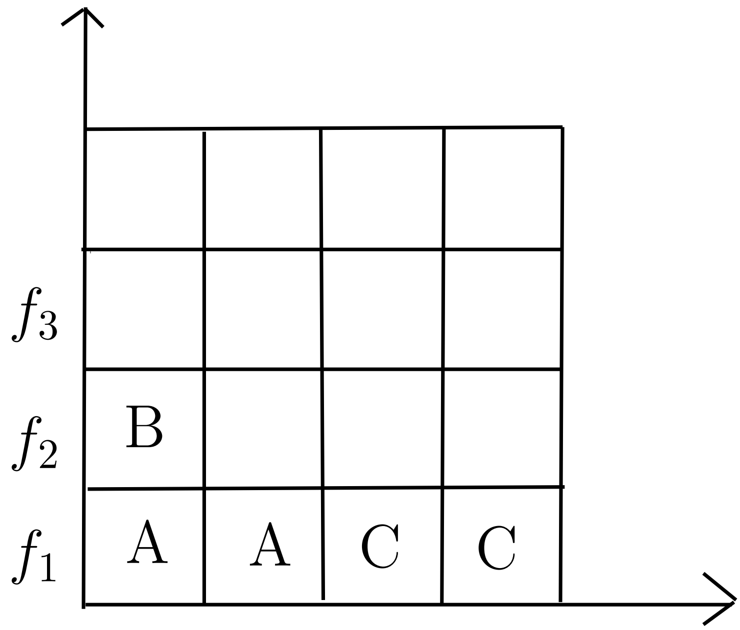

Figure 5

However note that if user C is given slots from subcarrier f_2 the 1 remaining needed slot (if it was assigned to subcarrier f_1) is equivalent to more than one (actually $\lceil 4/3 \rceil = 2$ slots, if assigned to subcarrier f_2).

Linear Programming formulation

The variables of the problem are x_{ij} : portion (percentage) of rate requirements of user i that are satisfied by carrier j, $i = 1, \ldots, K$ and $j = 1, \ldots, N$. The variable vector is $\mathbf{x} = (x_{ij} : i = 1, \ldots, K, j = 1, \ldots, N)$.

A problem instance is described by the long $NK \times 1$ vector

$\mathbf{a} = (a_{ij} : i = 1, \ldots, K, j = 1, \ldots, N)$. As mentioned before, the parameters a_{ij} are known to the BS for each user i and each subcarrier j and denote the number of time slots that are needed by user i to entirely fulfil its rate requirements when assigned *only* to carrier j. Let $a_{ij} \in \mathcal{R}$. Let all the subcarriers have capacity of C time slots per frame.

We want to minimize the total number of time slots that are needed to satisfy all users:

$$\min_{\mathbf{x}} \sum_{i=1}^{K} \sum_{j=1}^{N} a_{ij} x_{ij} \tag{72}$$

subject to the following constraints:

$$\sum_{i=1}^{K} a_{ij} x_{ij} \leq C, \text{ for } j = 1, \ldots, N. \tag{73}$$

and

$$\sum_{j=1}^{N} x_{ij} = 1, \text{ for } i = 1, \ldots, K. \tag{74}$$

The first constraint specifies that the available slot capacity must not be exceeded. The second constraint says that user rate requirements need to be satisfied. Also, for the variables x_{ij} it is $0 \leq x_{ij} \leq 1$. Since the objective function and the constraints are linear in the variable vector $\mathbf{x}$, the formulation above is an LP problem.

10 Duality

We now turn our attention to a very important topic of Linear Programming, that of duality. Duality appears in LP as well as in non-LP problems.

Consider the original LP optimization problem as we have seen it till now:

$$\min \mathbf{c}^T \mathbf{x} \tag{75}$$

subject to:

$$A\mathbf{x} = \mathbf{b}, \text{ and } \mathbf{x} \geq 0. \tag{76}$$

This is called *Primal* problem (P).

Suppose there exists an optimal solution $\mathbf{x}^*$. Matrix A has dimension $m \times n$ (that means that there are m constraints). We define $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)$ where λ_i denotes the additional cost per unit if constraint i is not fulfilled.
Then we have:

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}) &= \mathbf{c}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{b} - A\mathbf{x}) && (77) \\ &= \mathbf{c}^T \mathbf{x} + \sum_{i=1}^{m} \boldsymbol{\lambda}_i (\mathbf{b}_i - {a_i}^T \mathbf{x}) && (78) \end{aligned}$$

and we have the unconstrained problem:

$$\min \mathbf{c}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{b} - A\mathbf{x}) \tag{79}$$

subject to $\mathbf{x} \geq \mathbf{0}$. The term $\boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x})$ is the penalty associated with violating the constraints. Define $g(\boldsymbol{\lambda})$ the optimal cost for the relaxed problem as a function of $\boldsymbol{\lambda}$,

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{x}\geq\mathbf{0}} \ \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T \cdot (\mathbf{b} - A\mathbf{x}) \tag{80}$$

We have

$$g(\boldsymbol{\lambda}) \leq \min_{\mathbf{x}\geq\mathbf{0}, A\mathbf{x}=\mathbf{b}} \ \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x}) \tag{81}$$

since $\min_{\mathbf{x}\in\mathcal{A}} f(x) \leq \min_{\mathbf{x}\in\mathcal{B}} f(x)$ for $\mathcal{B} \subseteq \mathcal{A}$. Thus, we further get:

$$g(\boldsymbol{\lambda}) \leq \min_{\mathbf{x}\geq\mathbf{0}, A\mathbf{x}=\mathbf{b}} \ \mathbf{c}^T\mathbf{x} \tag{82}$$

or

$$g(\boldsymbol{\lambda}) \leq \mathbf{c}^T\mathbf{x}^* \tag{83}$$

since $\mathbf{x}^*$ is feasible solution for the primal problem.

For each $\boldsymbol{\lambda}$, $g(\boldsymbol{\lambda})$ is a lower bound on the optimal cost of the primal problem, $\mathbf{c}^T\mathbf{x}^*$. The question now in to find the best (highest) lower bound. This is the dual problem.

10.1 Dual Problem

The dual problem is a maximization problem, namely the maximation of the lower bound for the cost $g(\boldsymbol{\lambda}) \leq \mathbf{c}^T\mathbf{x}^*$.
The primal problem without the constraints is:

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{x}\geq\mathbf{0}} \mathbf{c}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{b} - A\mathbf{x}) \tag{84}$$

and then

$$g(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^T\mathbf{b} + \min_{\mathbf{x}\geq\mathbf{0}}(\mathbf{c}^T - \boldsymbol{\lambda}^T A)\mathbf{x} \tag{85}$$

Now,

$$\min_{\mathbf{x}\geq\mathbf{0}}(\mathbf{c}^T - \boldsymbol{\lambda}^T A)\mathbf{x} = 0, \text{ if } \mathbf{c}^T - \boldsymbol{\lambda}^T A \geq \mathbf{0}^T \tag{86}$$

else it is $-\infty$. Hence, in maximizing $g(\boldsymbol{\lambda})$, we must only consider those values of $\boldsymbol{\lambda}$ for which $g(\boldsymbol{\lambda})$ is *not* $-\infty$.

The dual problem is therefore:

$$\max \boldsymbol{\lambda}^T\mathbf{b} \tag{87}$$

subject to the constraints:

$$\boldsymbol{\lambda}^T A \leq \mathbf{c}^T \tag{88}$$

Notice that the dual has no constraints on the sign of $\boldsymbol{\lambda}$. The primal problem is a minimization problem, whereas the dual problem is a maximization problem. In the dual problem $\mathbf{c}^{\mathbf{T}}$ (the cost vector for the primal) has become right-hand side of the constraints and vector b has became the rate of benefit.

If we have an LP problem in inequality form (constraints are $A\mathbf{x} \geq \mathbf{b}$), we convert it to the standard form by using a slack variable $\mathbf{s} \geq \mathbf{0}$:

$$A\mathbf{x} - \mathbf{s} = \mathbf{b} \tag{89}$$

or

$$[A \ \ -I](\mathbf{x} \ \mathbf{s})^T = \mathbf{b} \tag{90}$$

Then, according to the previously found dual, we will have the dual constraints:

$$\boldsymbol{\lambda}^T[A \ \ -I] \leq [\mathbf{c}^T \ \ \mathbf{0}^T] \tag{91}$$

or

$$\boldsymbol{\lambda}^T A \leq \mathbf{c}^T \tag{92}$$

and

$$\boldsymbol{\lambda}^T(-I) \leq \mathbf{0} \Leftrightarrow \boldsymbol{\lambda} \geq \mathbf{0} \tag{93}$$

So, if in the primal problem the constraints are inequalities, in the dual, we have the constraint that the dual variables $\boldsymbol{\lambda} \geq \mathbf{0}$.

11 Primal LP problems and their dual problems

11.1 Forms of the primal problem

Below we show some primal problems of linear programming and their corresponding dual.

No	Primal Problem	Dual Problem
1	$\min \mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} = \mathbf{b}$ $\mathbf{x} \geq \mathbf{0}$	$\max \boldsymbol{\lambda}^T\mathbf{b}$ s.t. $\boldsymbol{\lambda}^T A \leq \mathbf{c}^T$ $\boldsymbol{\lambda}$ unrestricted
2	$\min \mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} = \mathbf{b}$	$\max \boldsymbol{\lambda}^T\mathbf{b}$ s.t. $\boldsymbol{\lambda}^T A = \mathbf{c}^T$ $\boldsymbol{\lambda}$ unrestricted
3	$\min \mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} \geq \mathbf{b}$	$\max \boldsymbol{\lambda}^T\mathbf{b}$ s.t. $\boldsymbol{\lambda}^T A = \mathbf{c}^T$ $\boldsymbol{\lambda} \geq \mathbf{0}$
4	$\min \mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} \geq \mathbf{b}$ $\mathbf{x} \geq \mathbf{0}$	$\max \boldsymbol{\lambda}^T\mathbf{b}$ s.t. $\boldsymbol{\lambda}^T A \leq \mathbf{c}^T$ $\boldsymbol{\lambda} \geq \mathbf{0}$

Generally, when the primal problem has inequality constraints, then in the dual we have the variables $\boldsymbol{\lambda} \geq \mathbf{0}$. When the primal problem has equality constraints, then in the dual the variables $\boldsymbol{\lambda}$ are unrestricted in sign.

Fact: The dual of the dual is the primal problem.

Proof: Assume the primal problem and its corresponding dual of equation (4) before. The dual problem can be written as

$$\min \boldsymbol{\lambda}^T(-\mathbf{b}) \qquad \text{s.t. } \boldsymbol{\lambda}^T(-A) \geq -\mathbf{c}^T,\ \boldsymbol{\lambda} \geq \mathbf{0} \tag{94}$$

The dual problem of the above is:

$$\max (-\mathbf{c})^T\mathbf{x} \qquad \text{s.t. } (-A)\mathbf{x} \leq (-\mathbf{b}),\ \mathbf{x} \geq \mathbf{0} \tag{95}$$

which can be written as

$$\min \mathbf{c}^T \mathbf{x} \qquad \text{s.t. } A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0} \tag{96}$$

which is the primal problem of equation (4). Thus, we proved that the dual problem of a dual problem is its primal problem.

11.2 Example (The Diet Problem)

A diet contains m different vitamins that need to be received daily with quantities at least equal to $b_1, ..., b_m$ respectively. The diet also have n different foods. Let a_{ij} denote the amount of vitamin i per unit of j-th food.

A company intends to propose a diet that is most economical. We can compose such a diet by choosing nonnegative food quantities $\mathbf{x} = (x_1, ...x_n)$. One unit quantity of food j and has a cost of c_j . We want to determine the cheapest diet that satisfies the nutritional requirements. This problem can be formulated as the LP primal problem,

$$\min \mathbf{c}^T \mathbf{x} \qquad \text{s.t. } A\mathbf{x} \geq \mathbf{0}, \ \mathbf{x} \geq \mathbf{0} \tag{97}$$

The corresponding dual problem can be defined as

$$\max \boldsymbol{\lambda}^T \mathbf{b} \qquad \text{s.t. } \boldsymbol{\lambda}^T A < \mathbf{c}^T, \ \boldsymbol{\lambda} \geq \mathbf{0} \tag{98}$$

where λ_i (dual problem variable) is the price of the unit quantity of vitamin $i = 1, ..., m$. In other words, this is the problem that another company (competitive to the first one) needs to solve. The company proposes a diet in which it has synthetically reproduced each food with vitamins. It then needs to find the pricing mechanism for each vitamin, i.e find the price vector $\boldsymbol{\lambda}$ to maximize its total benefit. At the same time, it has the constraints that the cost of the equivalent for food j should be,

$$\lambda_1 a_{1j} + \lambda_2 a_{2j} + ... + \lambda_m a_{mj} \leq c_j. \tag{99}$$

The inequality above should hold in order for the second company to be competitive. One unit quantity of food j has a production cost c_j and a price $\boldsymbol{\lambda}^T \mathbf{A}_j$ where $\mathbf{A}_j$ is the j-th column of matrix A, with elements a_{ij}, $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

11.3 Theorems and Lemmas in Duality

11.3.1 Duality Theorem

If the primal problem has an optimal solution $\mathbf{x}^*$, then so does the dual (it has an optimal solution $\boldsymbol{\lambda}^*$) and the optimal values of their respective objective functions are equal. In other words,

$$\mathbf{c}^T \mathbf{x}^* = \boldsymbol{\lambda}^{*T} \mathbf{b}. \tag{100}$$

Before coming to that, we will show that $\mathbf{c}^T \mathbf{x}^* \geq \lambda^T \mathbf{b}$.

11.3.2 Weak Duality Lemma

Suppose that $\mathbf{x_0}$ and λ_0 are feasible solutions to primal and dual problems respectively. Then,

$$\mathbf{c}^T\mathbf{x}_0 \geq \boldsymbol{\lambda}_0^T\mathbf{b}. \tag{101}$$

This inequality is known as the *Weak Duality Lemma*. Now, set $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_0$ and $\mathbf{x}^* = \mathbf{x}_0$ and we get

$$\mathbf{c}^T\mathbf{x}^* \geq \boldsymbol{\lambda}^{*T}\mathbf{b}. \tag{102}$$

Every feasible solution of the dual problem gives a lower bound on the value of the objective function of the primal problem. Also, every feasible solution to the primal problem gives an upper bound on the value of the objective function of the dual.

Note that the weak duality lemma holds for any of the four primal-dual pairs that we mentioned in the beginning (it can be verified easily).

Two immediate conclusions are:

- If the primal problem is unbounded so that $\mathbf{c}^T\mathbf{x}^* = -\infty$, then the dual problem is infeasible.
- If the dual problem is unbounded so that $\boldsymbol{\lambda}^*\mathbf{b} = +\infty$, then the primal problem is infeasible.

Note: By saying a problem is infeasible, it means that its set of feasible solutions is the empty set. Also, it is possible that both primal and dual problem be infeasible.

11.3.3 Theorem

Suppose that $\mathbf{x}$ and λ are feasible solutions to the primal and dual problem respectively. If $\mathbf{c}^T\mathbf{x} = \boldsymbol{\lambda}^T\mathbf{b}$, then $\mathbf{x}$ and $\boldsymbol{\lambda}$ are optimal solutions to the primal and dual problems respectively.

11.3.4 Lemma

Suppose that $\mathbf{x}$ and $\boldsymbol{\lambda}$ are feasible solutions to the primal and dual problem respectively. Then,

$$\mathbf{c}^T\mathbf{x} \geq \boldsymbol{\lambda}^T A\mathbf{x} \geq \boldsymbol{\lambda}^T\mathbf{b}. \tag{103}$$

Proof: For the first inequality, we must show that

$$(\mathbf{c}^T - \boldsymbol{\lambda}^T A)\mathbf{x} \geq 0, \tag{104}$$

which is true because $\mathbf{c}^T - \boldsymbol{\lambda}^T A \geq \mathbf{0}$ and $\mathbf{x} \geq \mathbf{0}$.
For the second inequality, we must show that

$$\boldsymbol{\lambda}^T(A\mathbf{x} - \mathbf{b}) \geq 0, \tag{105}$$

which is also true because $\boldsymbol{\lambda} \geq \mathbf{0}$ as a feasible solution to the dual problem and $A\mathbf{x} - \mathbf{b} \geq \mathbf{0}$ as $\mathbf{x}$ is a feasible solution to the primal problem.

If the constraint $A\mathbf{x} \geq \mathbf{b}$ was replaced by $A\mathbf{x} = \mathbf{b}$ then the respective dual problem would be

$$\max \boldsymbol{\lambda}^T\mathbf{b}, \qquad \text{s.t. } \boldsymbol{\lambda}^T A \leq \mathbf{c}^T, \ \boldsymbol{\lambda} \text{ unrestricted}, \tag{106}$$

which shows us that $\boldsymbol{\lambda}^T(A\mathbf{x} - \mathbf{b}) \geq 0$, is always true.

Note that, since solutions of primal and dual problem coincide, we can choose to solve the one that has the least complexity. This is of great importance especially in the case that one LP problem requires a large amount of resources in order to be solved (or maybe is inherently difficult to solve it) whereas the corresponting dual problem requires significantly less amount of recources. Also, dual problem help us to interpret the characteristics of LP problem and solve some of the problems in a distributed way.

11.3.5 Strong Duality Theorem

Suppose that $\mathbf{x}^*$ and $\boldsymbol{\lambda}^*$ are feasible solutions to the primal and dual problem respectively and $\mathbf{c}^T\mathbf{x}^* \geq \boldsymbol{\lambda}^{*T}\mathbf{b}$. There are four options about the solutions of primal and dual problems:

- Both problems have optimal solutions (of finite value).
- Both problems are infeasible (their sets of feasible solutions are empty).
- The primal problem is unbounded and the dual problem is infeasible.
- The primal problem is infeasible and the dual problem is unbounded.

11.3.6 Theorem: Complementary Slackness Conditions

The feasible solutions $\mathbf{x}^*$ and λ^* to the primal and dual problem respectively are optimal solutions if and only if

1. $(\mathbf{c}^T - \boldsymbol{\lambda}^{*T} A)\mathbf{x}^* = 0$
2. $\boldsymbol{\lambda}^{*T}(A\mathbf{x}^* - \mathbf{b}) = 0.$

We omit the proofs and focus on their interpretation.

1. We know that $\mathbf{x}^* \geq \mathbf{0}$ and $\mathbf{c}^T - \boldsymbol{\lambda}^{*T} A \geq \mathbf{0}^T$. This means that,

$$(c_j - \boldsymbol{\lambda}^{*T}\mathbf{A}_j)x_j^* = 0 \;\forall\; j = 1, ..., n. \tag{107}$$

The conclusions are:

$$\text{If } x_j^* > 0 \Rightarrow \boldsymbol{\lambda}^{*T}\mathbf{A}_j = c_j \tag{108}$$

$$\text{If } \boldsymbol{\lambda}^{*T}\mathbf{A}_j < c_j \Rightarrow x_j^* = 0, \tag{109}$$

where c_j is the j-th element of vector $\mathbf{c}$, x_j^* is the j-th element of vector $\mathbf{x}^*$ and $\mathbf{A}_j$ is the j-th column of matrix A.

Thus, if a component of the primal solution is strictly positive, the corresponding constraint in the dual must be met with equality at the optimal solution. And also, if an inequality constraint at the dual is not met with ”clean” inequality at the optimal solution, the corresponding variable at the primal optimal solution is zero.

2. We know that $\boldsymbol{\lambda}^* \geq 0$ and $A\mathbf{x}^* \geq \mathbf{b}$. This means that

$$\lambda_i^{*T}(\mathbf{a}_i^T\mathbf{x}^* - b_i) = 0 \;\forall\; i = 1, ..., m. \tag{110}$$

The conclusions are:

$$\text{If } \lambda_i^* > 0 \Rightarrow \mathbf{a}_i^T \mathbf{x}^* = b_i \tag{111}$$

$$\text{If } \mathbf{a}_i^T \mathbf{x}^* > b_i \Rightarrow \lambda_i^* = 0. \tag{112}$$

where λ_i is the i-th element of vector λ, $\mathbf{a}_i$ is the i-th row of matrix A and b_i is the i-th element of vector $\mathbf{b}$. Similar statements can be made here as those for case 1.

So, in general, we have:

- If the variable of (P) > 0 then constraint of (D) is equality
- If the variable of (D) > 0 then constraint of (P) is equality
- If constraint of (P) is inequality then the variable of (D) $= 0$
- If constraint of (D) is inequality then the variable of (P) $= 0$

where the Primal problem is denoted as (P) and the dual problem as (D).

12 Interpretation of dual variables

Consider the primal LP problem

$$\min \mathbf{c}^T \mathbf{x} \tag{113}$$

subject to:

$$A\mathbf{x} = \mathbf{b}, \ \mathbf{x} \geq \mathbf{0} \tag{114}$$

Recall as r_i the reduced cost coefficients from the simplex algorithm. Let the optimal BFS be $\mathbf{x} = (\mathbf{x}_B \ \mathbf{0})$. At the end of the simplex algorithm, matrix A is partitioned as $[B \ D]$, where B is the matrix of linearly independent columns, corresponding to the basis. As we know, $\mathbf{x}_B = B^{-1}\mathbf{b}$. Recall also that at the end of the simplex algorithm the vector of relative coefficients corresponding to the non-basic variables is $\mathbf{r}^T \geq \mathbf{0}$ (a (n-m) dimension vector). It can be easily shown that at the end of the simplex algorithm, it is

$$\mathbf{r}^T = \mathbf{c}_D^T - \mathbf{c}_B^T B^{-1} D, \tag{115}$$

where for the cost vector is separated in two parts corresponding to the basic and non-basic variables, as

$$\mathbf{c} = (\mathbf{c}_B \ \mathbf{c}_D). \tag{116}$$

Since at the optimal solution it is $\mathbf{r}^T \geq \mathbf{0}^T$, we have the inequality

$$\mathbf{c}_D^T - \mathbf{c}_B^T B^{-1} D \geq 0 \Rightarrow \mathbf{c}_B^T B^{-1} D \leq \mathbf{c}_D^T. \tag{117}$$

Now, consider the dual problem,

$$\max \boldsymbol{\lambda}^T \mathbf{b} \tag{118}$$

subject to:

$$\boldsymbol{\lambda}^T A \leq \mathbf{c}^T, \tag{119}$$

and $\boldsymbol{\lambda}$ unrestricted in sign. Define $\boldsymbol{\lambda} = \mathbf{c}_B^T B^{-1}$. We are going to prove that $\boldsymbol{\lambda}^T = \mathbf{c}_B^T B^{-1}$ is the optimal solution to the dual problem.

First, we will check if $\boldsymbol{\lambda}$ is a feasible solution to the dual problem, i.e. if it satisfies $\boldsymbol{\lambda}^T A \leq \mathbf{c}^T$. Indeed, it is $\boldsymbol{\lambda}^T A = \boldsymbol{\lambda}^T [B\ D] = [\boldsymbol{\lambda}^T B\ \boldsymbol{\lambda}^T D] = [\mathbf{c}_B^T\ \mathbf{c}_B^T B^{-1} D] \leq [\mathbf{c}_B^T\ \mathbf{c}_D^T] = \mathbf{c}^T$. Thus, $\boldsymbol{\lambda}$ is a feasible solution of the dual.

Next, we prove that $\boldsymbol{\lambda}$ is the optimal solution for the dual problem. We have $\boldsymbol{\lambda}^T \mathbf{b} = \mathbf{c}_B^T B^{-1} \mathbf{b} = \mathbf{c}_B^T \mathbf{x}_B = \mathbf{c}^T \mathbf{x}$, where $\mathbf{x_B}$ is the vector of the basic variables. Therefore, we proved that if the primal LP has an optimal BFS with basis B, then $\lambda^T = \mathbf{c}_B^T B^{-1}$ is optimal solution for the dual problem. Vector $\boldsymbol{\lambda}^T$ is called vector of *simplex multipliers*.

12.1 Sensitivity Analysis

Consider the primal problem:

$$\begin{aligned} &\min \mathbf{c}^T \mathbf{x} \\ \text{subject to: } &A\mathbf{x} = \mathbf{b} \\ &\mathbf{x} \geq \mathbf{0} \end{aligned}$$

Let the optimal basic feasible solution (BFS) be $\mathbf{x} = (\mathbf{x_B}, 0)$, with basis matrix B and $\mathbf{x}_B = B^{-1}\mathbf{b}$.

Let's assume that the right-hand side of the constraints changes by a small $\Delta\mathbf{b}$, that is $\mathbf{b} \rightsquigarrow \mathbf{b} + \Delta\mathbf{b}$. For small changes $\Delta\mathbf{b}$, the basis matrix B does not change. After the change, the new optimal solution will be

$$\mathbf{x}' = (\mathbf{x}_B{}', 0)$$

where

$$\mathbf{x}_B{}' = B^{-1}(\mathbf{b} + \Delta\mathbf{b}) = \mathbf{x}_B + \Delta\mathbf{x}_B$$

since $\Delta\mathbf{x}_B = B^{-1}\Delta\mathbf{b}$.

The value of the objective function before the change was $\mathbf{c}_B^T\mathbf{x}_B$. After the change, it is $\mathbf{c}_B^T(\mathbf{x}_B + \Delta\mathbf{x}_B)$.
The objective function has been changed by quantity

$$\Delta z = \mathbf{c}_B^T\Delta\mathbf{x}_B = \mathbf{c}_B^T B^{-1}\Delta\mathbf{b} = \boldsymbol{\lambda}^T\Delta\mathbf{b}$$

Suppose there was only one constraint, $m = 1$, then: $\Delta z = \lambda\Delta b \Rightarrow \lambda = \frac{\Delta z}{\Delta b}$. So, λ can be interpreted as the rate of change of the value of the objective function for small changes of the constraints. Since the constraints often represent resources, λ can be interpreted as the *price* per unit of the resources. This is also obvious from the fact that $\Delta z = \lambda\Delta b$ if Δz is the profit delivered by quantity b of the resources. Therefore, λ specifies the change in the value of the objective function with respect to a unit change in resources. It is also called *shadow price* or *marginal cost*.

For more than one constraint, $m > 1$, it is $\lambda_i = \dfrac{\partial z}{\partial b_i}$ and λ_i is the rate of change in the value of the objective function with respect to a change in constraint i.

Note that the more the value of λ increases, the more the cost becomes sensitive to changes.

Next, we will study some examples of LP problems and will try to interpret duality theorems for them.

12.2 Shortest Path Problem

First, we consider the shortest path (SP) problem. This problem is a special case of the more general *minimum cost flow problem*.

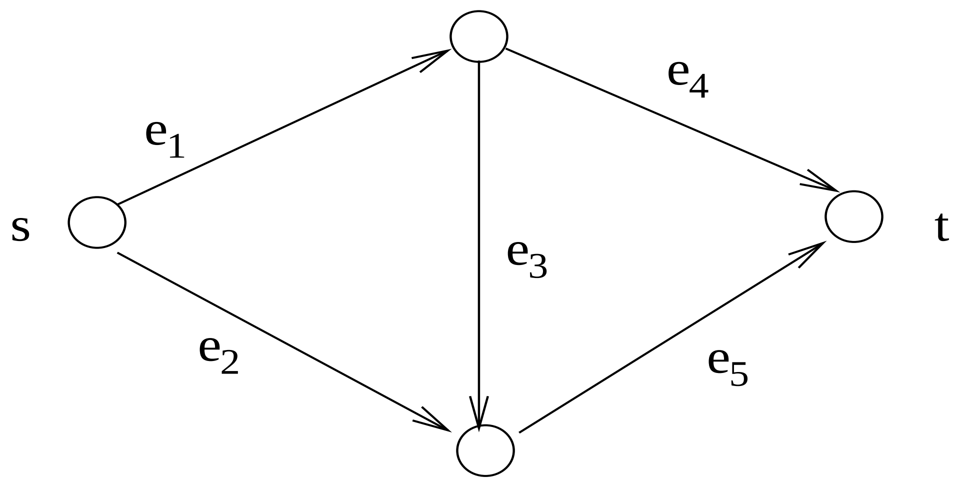

Figure 9: An example network.

Consider a directed graph $G = (V, E)$, where V represents the set of nodes and E represents the set of edges of the graph. Let $c_j \geq 0$ be the cost associated with each edge $e_j \in E$. The min-cost path problem is the problem of finding a directed path of minimum total cost from a source node s to a destination node t. For the special case where $c_j = 1$ for each $e_j \in E$, we have the shortest path problem, i.e the problem of finding the shortest route to th destination.

The problem arises in several applications such as routing, power control etc. The cost of an edge (i, j) may denote the required power to reach from transmitter i to receiver j. Then, the shortest path specifies the route with the minimum total power consumption. energy costs can also be similarly incorporated in that context. Costs may also denote delays in packet forwarding in a network (which may model link rates or queueing delays at nodes).

In the LP problem, the feasible set is
$\mathcal{F}$ = sequences $\{P = (e_{j_1}, \ldots, e_{j_k})\}$ such that the sequence is a directed path from s to t, i.e all possible paths leading from source to destination.

Let the path cost be $c(P) = \sum_{i=1}^{k} c_{j_i}$. Define Let the *node-edge incidence matrix* $A = [A_{ij}], i = 1, ..., |V|, j = 1, ..., |E|$, with

$$A_{ij} = \begin{cases} +1 & \text{if edge } e_j \text{ leaves node } i \\ -1 & \text{if edge } e_j \text{ enters node } i \\ 0 & \text{otherwise.} \end{cases}$$

Example: For the graph of Figure (9) , it is

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

The rows, starting from the first one, stand for nodes s, t, a, b respectively. The columns, starting from the first one, stand for edges e_1, e_2, e_3, e_4, e_5 respectively.

Associate a flow variable f_j with each edge e_j to represent flow of an imaginary fluid through e_j. Consider the flow vector $\mathbf{f} = (f_j : j = 1, ..., |E|)$. The flow conservation principle at each node i can be expressed as the equation

$$\mathbf{a}_i^T \mathbf{f} = 0, \; i \neq \{s, t\}, \tag{120}$$

where $\mathbf{a}_i$ is the i-th row of matrix A. A path from s to t is a flow of one unit leaving s and entering t. this flow satisfies the flow conservation equations above at each intermediate node in the path, and also $\mathbf{a}_s^T \mathbf{f} = +1$ and $\mathbf{a}_t^T \mathbf{f} = -1$. Overall, the constraints are written as

$$A\mathbf{f} = \begin{bmatrix} +1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The first two rows stand for s and t respectively and the next rows stand for nodes $i \neq s, t$. The primal problem can be stated as:

$$\min \mathbf{c}^T \mathbf{f}$$
$$\text{subject to: } \mathbf{f} \geq \mathbf{0}$$

$$A\mathbf{f} = \begin{bmatrix} +1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

In its most general form, the problem is the minimum cost flow problem that has solutions $\mathbf{f} \geq \mathbf{0}$ and $\mathbf{f} \leq \mathbf{1}$. The shortest path problem is a special case of the minimum cost flow problem, where $f_e \in \{0, 1\}$. At the optimal solution,

- if $f_{e_j} = 1$, then edge e_j is part of the optimal path P^* to the destination.
- if $f_{e_j} = 0$, then edge e_j is not path of the optimal path.

In the dual problem there is one variable for each node in the network. The dual problem is:

$$\max (\lambda_s - \lambda_t)$$
$$\text{subject to : } \lambda_i - \lambda_j \leq c_{ij} \text{for each edge} \quad e = (i, j)$$
$$\lambda_i \text{ unrestricted in sign}$$

The constraints can be seeing as emerging from the dual constraints $\boldsymbol{\lambda}^T A \leq \mathbf{c}^T$. Let λ_i be the cost of having one flow unit at node i. The complementary slackness conditions for this problem in the optimal solution are written as

$$(\lambda_i - \lambda_j - c_{ij})\, f_{ij} = 0 \tag{121}$$

If $\lambda_i - \lambda_j < c_{ij} \Rightarrow f_{ij} = 0$. This means that if the cost of transporting one unit of flow from node i to node j is more than the difference in costs of having the flows at i and j, then edge (i, j) is not included in the shortest path (because there can be another way of transporting one unit of flow from i to j. On the other hand, if $f_{ij} > 0 \Rightarrow \lambda_i - \lambda_j = c_{ij}$, and this means that edge (i, j) is included in the shortest path.

For the shortest path problem from a single source to a single destination and non-negative edge costs, there is the Dijkstra algorithm. A more general algorithm for multiple sources and destinations and also negative costs is the Bellman-Ford algorithm.

12.3 Assignment Problem

Consider the following problem. There exist n tasks / jobs to be assigned to n persons. The benefit of assigning task j to person i is a_{ij}. Alternatively, a_{ij} denotes the cost of assigning task j to person i. Depending on one or the other case, we have the min-cost or max-weight assignment problem. We will consider the second case.

Define the variables

$$x_{ij} = \begin{cases} 0 & \text{if task } j \text{ is not assigned to person } i \\ 1 & \text{if task } j \text{ is assigned to person } i \end{cases} \tag{122}$$

The maximum weight assignment problem (P) is the following:

$$\max_{\{x_{ij}\}} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_{ij} \tag{123}$$

subject to:

$$\sum_{i=1}^{n} x_{ij} = 1, \ \forall \text{ task } j \tag{124}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \ \forall \text{ person } i, \tag{125}$$

and $x_{ij} \in \{0, 1\}$.

Define a dual variable λ_j for each constraint corresponding to a task j and a dual variable μ_i for each constraint corresponding to a person i. These can denote the cost of having the task j assigned and the cost of having occupied person i. The dual problem is:

$$\min \sum_{j=1}^{n} \lambda_j + \sum_{i=1}^{n} \mu_i \tag{126}$$

subject to:

$$\lambda_j + \mu_i \geq a_{ij}, \forall (i,j) \tag{127}$$

and $\{\lambda_j\}$, $\{\mu_i\}$ unrestricted in sign.

Based on complementary slackness, we have at the optimal solution: $(a_{ij} - \lambda_j - \mu_i) x_{ij} = 0$. Thus, if $\lambda_j + \mu_i > a_{ij} \Rightarrow x_{ij} = 0$. This means that if the benefit a_{ij} of assigning task j to person i is less than the incurred cost, then do not assign task j to person i. On the other hand, if $x_{ij} > 0 \Rightarrow \lambda_j + \mu_i = a_{ij}$, i.e it is valid and meaningful to assign task j to person i if the incurred benefit equals the incurred cost.

12.4 Minimum Cost Flow Problem

Consider a network represented by a directed graph $G(V,E)$. Let c_{ij} be the cost of transferring one unit of flow through the edge (i,j), and u_{ij} be the capacity of edge (i,j), i.e the maximum flow that can be transported through the edge (i,j). Also define

$$b_i \begin{cases} > 0 & \text{if } i \text{ is the source} \\ < 0 & \text{if } i \text{ is the destination} \\ = 0 & \text{otherwise} \end{cases} \quad \text{For each node}$$

in the network, we have

$$b_i + \sum_{j:(j,i)\in E} f_{ji} = \sum_{j:(i,j)\in E} f_{ij}, \tag{128}$$

namely the flow conservation equation. Also, $0 \leq f_{ij} \leq u_{ij}$. The minimum cost flow problem is:

$$\min \sum_{(i,j)\in E} c_{ij} f_{ij}, \tag{129}$$

subject to the constraints above. The problem is called uncapacitated min-cost flow problem, if $u_{ij} = +\infty$ for all edges (i,j), otherwise it is called capacitated.