## The flow problem as a LP problem

We let $x_e$ be the flow on edge $e$. We have the constraints $0 \leq x_e \leq c(e)$ for all $e$. For each node $x$ except $s$ and $t$ we have

$$\sum_{e \in In(x)} x_e = \sum_{e \in Ut(x)} x_e$$

We set

$$v = \sum_{e \in Ut(s)} x_e$$

The flow problem can be written as

Maximize $v$

when

$$\begin{cases} v = \sum_{e \in Ut(s)} x_e & \\ \sum_{e \in In(x)} x_e = \sum_{e \in Ut(x)} x_e & \text{for all x except s,t} \\ 0 \leq x_e \leq c(e) & \text{for all edges} \end{cases}$$

## A transport problem

The company Carla produces milk in 4 different plants. The milk is delivered to 5 customers. Carla has to consider three things:

1. The capacities of the plants.
2. The demands of the customers.
3. The costs of the transports between plants and customers.

Let us call the plants F1, F2, F3, F4.

Capacity:

| F1 | F2 | F3 | F4 |
|---|---|---|---|
| 30 | 40 | 30 | 40 |

(The numbers represent 1000 liters.)

Let us call the customers K1, K2, K3, K4, K5.

Demand:

| K1 | K2 | K3 | K4 | K5 |
|---|---|---|---|---|
| 20 | 30 | 15 | 25 | 20 |

(The numbers represent 1000 liters.)

Transport costs:

| | K1 | K2 | K3 | K4 | K4 |
|---|---|---|---|---|---|
| F1 | $2,80$ | $2,55$ | $3,25$ | $4,30$ | $4,35$ |
| F2 | $4,30$ | $3,15$ | $2,55$ | $3,30$ | $3,50$ |
| F3 | $3,00$ | $3,30$ | $2,90$ | $4,30$ | $3,40$ |
| F4 | $5,20$ | $4,45$ | $3,50$ | $3,75$ | $2,45$ |

Goal:

Decide how the ''flow'' to the customers should be so that

1. The customers are satisfied.
2. The cost are minimal.

Mathematical model:

Use variables $x_{ij}$ for the flow from plant $i$ to customer $j$.

What demands do we have?

1. Capacities

Ex: For plant 1 we should have

$$x_{11} + x_{12} + x_{13} + x_{14} + x_{15} \leq 30000$$

2. Demand

Ex: For customer 1 we should have
$x_{11} + x_{21} + x_{31} + x_{41} = 20000$

Cost:

$$z = 2,80x_{11} + 2,55x_{12} + ... + 2,45x_{45}$$

We use the following definitions:

Let $c_{ij}$ be the cost for transport from plant $i$ to customer $j$.

Let $s_i$ be the capacity for plant $i$.

Let $d_j$ be the demand of customer $j$.

The problem can now be written as

Minimize $\sum_{i=1}^{4} \sum_{j=1}^{5} c_{ij} x_{ij}$

when

$$\begin{aligned} \sum_{j=1}^{5} x_{\text{ij}} \leqslant s_i \;\; i = 1, 2, 3, 4 \\ \sum_{i=1}^{4} x_{\text{ij}} = d_j \;\; j = 1, 2, 3, 4, 5 \\ x_{\text{ij}} \geqslant 0 \end{aligned}$$

## Linear Programming

A Linear Programming problem is the following:

Input: We have $n$ variables $x_1, x_2, ..., x_n$ and $m$ linear equalities and/or inequalities in the variables. We can also have constraints that say that some (all) of the variables should be non-negative. We are given a linear function $f$ in the variables.

Goal: We want to find values for the variables so that the constraints are fulfilled and the function $f$ is optimized (maximized/minimized).

## Different forms

We can express an LP-problem on different forms. We have

1. General form: That is the one described above.

2. Canonical form: Essentially a form with just inequalities. This form is suitable for analyzing mathematical properties of solutions.

3. Standard form: Essentially a form with just equalities. This form is used when actually finding solutions.

The general form covers all LP-problems. But all problems can in a certain way be translated to equivalent problems on canonical and standard forms.

## Canonical Form

A linear programming problem on canonical form is

Minimize $\sum_{j=1}^{n} c_j x_j$

when
$\sum_{j=1}^{n} a_{\mathsf{ij}} x_j \leqslant b_i \; i = 1, 2, \ldots, m$
$x_j \geqslant 0$

In some texts the authors use maximization instead of minimization. This doesn't matter much since we can always translate one form to the other by changing the sign of the $c_i$ :s.

## Translations

If we have a problem that is not on canonical form we can rewrite it on canonical form. We show how it can be done by looking at some examples:

**Example:**

Minimize

$$x_1 + 2x_2 - x_3$$

when

$$\begin{cases} x_1 + x_3 = 1 \\ x_2 - x_3 \geqslant 3 \end{cases}$$

Inequalities "in the wrong direction"can be turned right by a sign change.

Equalities can be turned into inequalities by using two using two inequalities for each equality.

In our problem we get

Minimize

$$x_1 + 2x_2 - x_3$$

when

$$\begin{cases} x_1 + x_3 \leqslant 1 \\ -x_1 - x_3 \leqslant -1 \\ x_3 - x_2 \leqslant -3 \end{cases}$$

## Towards solutions: Standard forms

Preparation: We transform the problem to so called standard form.

Standard form: We have equalities instead of inequalities.

Ex:

Minimize $z = 3x_1 + 5x_2 - x_3$

when

$$x_1 - x_2 + 2x_3 = 5$$
$$x_1 + 2x_2 + 4x_3 = 12$$
$$x_1, x_2, x_3 \geqslant 0$$

We get equalities by introducing Slack Variables.

Ex:
Let us assume that we have the inequality
$x_1 + 3x_2 \leqslant 10$

We set $x_3 = 10 - (x_1 + 3x_2)$

$x_3$ is a new slack variable.

We get the equality $x_1 + 3x_2 + x_3 = 10$

## The Simplex Method

There is a famous algorithm called the Simplex Algorithm that solves these problems. We will describe this algorithm without going to much into details.

Preparation: We transform the problem to so called standard form

Standard form: We have equalities instead of inequalities.

Ex:

Minimize $z = 3x_1 + 5x_2$ - $x_3$

when

$x_1 - x_2 + 2x_3 = 5$
$x_1 + 2x_2 + 4x_3 = 12$
$x_1, x_2, x_3 \geqslant 0$

We get equalities by introducing Slack Variables.

Ex: Let us assume that we have the inequality $x_1 + 3x_2 \leqslant 10$

We set $x_3 = 10$- $(x_1 + 3x_2)$

We get the equality $x_1 + 3x_2 + x_3 = 10$

Let us assume that we have the following problem:

Maximize $z = 20x_1 + 18x_2$

when

$x_1 + 10x_2 \leq 3600$

$16x_1 + 12x_2 \leq 5400$

$x_1, x_2 \geq 0$

$7x_1 + 10x_2 \leqslant 3600$ reduces to $7x_1 + 10x_2 + x_3 = 3600$

$16x_1 + 12x_2 \leqslant 5400$ reduces to $16x_1 + 12x_2 + x_4 = 5400$

We get

Maximize $\quad z = 20x_1 + 18x_2$

when

$7x_1 + 10x_2 + x_3 = 3600$

$16x_1 + 12x_2 + x_4 = 5400$

$x_1, x_2, x_3, x_4 \geqslant 0$

**Standard form**

Minimize $z = \sum_{j=1}^{n} c_j x_j$

when
$\sum_{j=1}^{n} a_{ij} x_j = b_i \ i = 1, ..., m$
$x_j \geqslant 0 \ j = 1, ..., n$

We can use matrix notation

Minimize $\bar{c}^T \bar{x}$

when
$A\bar{x} = \bar{b}$
$\bar{x} \geqslant \bar{0}$

Our previous example will look like:

$$A = \begin{pmatrix} 7 & 10 & 1 & 0 \\ 16 & 2 & 0 & 1 \end{pmatrix}$$

$$\bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \; \bar{c} = \begin{pmatrix} -20 \\ -18 \\ 0 \\ 0 \end{pmatrix} \; \bar{b} = \begin{pmatrix} 3600 \\ 5400 \end{pmatrix}$$

Minimize (-20 -18 0 0 ) $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$

when

$$\begin{pmatrix} 7 & 10 & 1 & 0 \\ 16 & 2 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3600 \\ 5400 \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \geqslant \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

## How to find a solution

Maximize $z = 20x_1 + 18x_2$.
When

$7x_1 + 10x_2 + x_3 = 3600$
$16x_1 + 12x_2 + x_4 = 5400$

How do we find the best solution?

One possibility is $x_3 = x_4 = 0$

$7x_1 + 10x_2 = 3600$
$16x_1 + 12x_2 = 5400$

If we solve the system we get $x_1 \approx 142$ , $x_2 \approx 260$

It gives us z $\approx$ 7520

But instead, we can put $x_2 = x_4 = 0$

We get the equations

$7x_1 + x_3 = 3600$
$16x_1 = 5400$

They give us $x_1 \approx 337$ $x_3 \approx 1237$

Then z $\approx$ 2362.

Are there more solutions?

**Basic solution:**

Let us assume that we have $n$ variables and $m$ equations. We also assume that all equations are linearly independent. We us say that we have set $n - m$ of the variables to 0.
Then the other $m$ variables have unique values. This gives us a basic solution .

**Feasible basic solution:**

If all variables are $\geqslant 0$ we have a feasible basic solution.

The solution to a LP-problem is always a feasible basic solution (FBS).

But which FBS?

**Method:**

Variables which are 0 (at a certain stage) are called non-basic variables. The other variables are called basic variables.

We test different FBS:s by changing the basic variables one at a time.

Ex: Minimize $z = 2x_1 + x_2$

when
$x_1 + x_2 = 10$
$x_1, x_2 \geqslant 0$

Set $x_1 = 0$.

Then $x_2 = 10$.

Then $z = 10$.

We now change basic variables so that $x_2 = 0$ .

Then $x_1 \approx 3{,}33$

we get z $\approx$ 6,67.

So we have found a better solution.

How do you know if you have found the best solution?

We look at our previous example:

$x_1 = 142$ $x_2 = 260$ $z = 7520$

Is that the best solution?

We can write

$x_3$ = 3600 - 7$x_1$ -10$x_2$
$x_4$ = 5400 - 16$x_1$ - 12$x_2$

$x_1$ = 0,158$x_3$ - 0,132$x_4$ + 142,1
$x_2$ = -0,2$x_3$ + 0,092$x_4$ + 260,5

That gives us $z$ = 20$x_1$ + 18$x_2$ = 20(0,15$x_3$ - 0,13$x_4$ + 142,1) + 18( -0,21$x_3$ + 0,09$x_4$ + 260,5) =
7520 - 0,62$x_3$ - 0,98$x_4$

Now we see that we would gain nothing by increasing $x_3$ or $x_4$.
We see that any change from this solution must end in a worse solution.

## General description of the Simplex Method

Let's say that we have a maximization problem and a FBS with basic variables $y_1$, $y_2$, ... , $y_m$ and non-basic variables $v_1$, $v_2$ , ... , $v_{n-m}$.

This means that $v_1 = v_2 = ... = v_{n-m} = 0$

We can then write $y_1$, $y_2$, ... , $y_m$ as functions of $v_1$, $v_2$ , ... , $v_{n-m}$

$y_1 = f_1\,(v_1, ..., v_{n-m})$
$y_2 = f_2\,(v_1, ..., v_{n-m})$
...
In the same way we can write $z$ as

$$z = c_1 v_1 + c_2 v_2 + ... + c_{n-m} v_{n-m} + z_0$$

If all $c_i$ are $< 0$ we must have an optimal solution.

If any $c_i > 0$, say $c_1 > 0$, we can increase z by increasing $v_1$. But then the values of the y:s must change. How much do they change?

We can increase $v_1$ until $f_k(v_1, v_2, ...) = 0$ for some $k$. Then $v_1$ will be a new basic variable and $y_k$ will be a new non-basic variable.
We go on like this until all $c_i \leq 0$. Then we have found the optimal solution.

If we have a minimization problem we must try to increase variables with $c_i < 0$. When all $c_i \geq 0$ we have a solution.

Ex:

Minimize $z = 2x_1 + 2x_2 + x_3$

when

$x_1 + x_2 + x_3 = 5$
$x_1 - x_2 + 2x_3 = 8$
$x_1, x_2, x_3 \geqslant 0$

One FBS is $x_2 = 0$ (non-basic variable).

We get

$x_1 + x_3 = 5 - x_2$
$x_1 + 2x_3 = 8 + x_2$

$x_1 = 2 - 3x_2$
$x_3 = 3 + 2x_2$

$z = 2(2 - 3x_2) + 2x_2 + (3 + 2x_2) = 7 - 2x_2$

We can increase $x_2$. But how much?

$x_1$ and $x_3$ must be $\geqslant 0$.

$x_1 = 2 - 3x_2$

This means $x_2 \leqslant \frac{2}{3}$

$x_3 = 3 + 2x_2$

This gives us no bound on $x_2$.

So $x_2 = \frac{2}{3}$ and $x_1 = 0$.

$x_3 = \frac{13}{3}$

We now write $x_2$, $x_3$ as functions of $x_1$.

$$x_2 = \frac{2}{3} - \frac{x_1}{3}$$

$$x_3 = 3 + 2x_2 = 3 + 2(\frac{2}{3} - \frac{x_1}{3}) = \frac{13}{3} - \frac{2x_1}{3}$$

$$z = 7 - 2x_2 = 7 - 2(\frac{2}{3} - \frac{x_1}{3}) = \frac{17}{3} + \frac{2x_1}{3}$$

Since we gain nothing by increasing $x_1$, we are done.

This is however far from the full story. There is a problem called degeneracy that can occur. This happens when when we have no $c_i > 0$ and some $c_i = 0$ (if we assume that we have a minimization problem). In that case we will have to chose some $i$ with $c_i = 0$. Then there is a chance that we could get into an infinite cycle. In practice, there are several ways to avoid this. Another problem is how to find a starting point for the algorithm. It turns out that we can use a modified variant of the simplex algorithm to solve this problem.

Actually, in worst case, the Simplex Algorithm is not a polynomial time algorithm. In practice, however, it is always considered efficient enough to be used.

## Dual Problems

For each LP problem we can give a so called dual problem.

Ex: We have the problem

Maximize $5x_1 + 2x_2$

when

$x_1 + x_2 \leqslant 10$

$2x_1 + 3x_2 \leqslant 20$

$x_1 , x_2 , x_3 \geqslant 0$

The dual problem is

Minimize $10v_1 + 20v_2$

when

$v_1 + 2v_2 \geqslant 5$

$v_1 + 3v_2 \geqslant 2$

$v_1, v_2 \geqslant 0$

How do we define the dual problem?

## Definition of dual problems

We write the problem on the form

Maximize $\bar{c}^T \bar{x}$

when
$A\bar{x} \leqslant \bar{b}$
$\bar{x} \geqslant \bar{0}$

The dual problem is

Minimize $\bar{b}^T \bar{v}$

when
$A^T \ \bar{v} \geqslant \bar{c}$
$\bar{v} \geqslant \bar{0}$

## The Duality Theorem

Let $P_1$ and $P_2$ be two dual problems. If one of the problems has a unique solution with value $M$, then the other problem also has a unique solution with value $M$. If we solve one of the problems we also get a solution to the other.

Our previous example:

We want to

Maximize $20x_1 + 18x_2$

when
$7x_1 + 10x_2 \leqslant 3600$
$16x_1 + 12x_2 \leqslant 5400$
$x_1, x_2 \geqslant 0$

The corresponding dual problem is

Minimize $3600v_1 + 5400v_2$

when

$7v_1 + 16v_2 \geqslant 20$
$10v_1 + 12v_2 \geqslant 18$
$v_1,\, v_2 \geqslant 0$

Both problems have the same value as solution.

## About solutions to LP problems

If we try to solve a LP problem three cases can occur.

1. The problem has a unique solution.

2. The inequalities defining the problem cannot be satisfied.

Ex: Minimize $x_1 + x_2$

when

$x_1$ - 2 $x_2$ $\leqslant$ -2

$x_1 + 3x_2 \leqslant 1$

$x_1$, $x_2$ $\geqslant$0

Then, of course, there is no solution to the problem.

3. The value can be arbitrarily large/small.

Ex: Maximize $x_1$ - $x_2$

when
$x_1 + x_2 \geqslant 10$
$x_1, x_2 \geqslant 0$
The problem is that $x_1 - x_2$ can be arbitrarily large. There is no solution.

Part of what the Duality Theorem tells us is that if one of two dual problems is of type 1, the other one must be of type 1 as well.

## Reduction of Shortest Path to Linear Programming

Find the shortest path $s \to t$ in a weighted (undirected) graph $G$.

We can define variables $x_e$ (one for each edge). We can the see that this LP-problems solves the Shortest Path-problem:

Minimize

$$\sum_e x_e w(e)$$

when

$$\begin{cases} 1 = \sum_{e \in Ut(s)} x_e \\ 1 = \sum_{e \in In(t)} x_e \\ \sum_{e \in In(x)} x_e = \sum_{e \in Ut(x)} x_e & \text{for all x except s,t} \\ 0 \leq x_e \text{ for all edges} \end{cases}$$

## The dual form

Dualization will give us variables $d_v$ (one for each node). Without going into details we state the dual form:

Maximize $d_t$

when

$$\begin{cases} d_u \leq d_v + w(u,v) & \text{for all edges } (u,v) \\ d_s = 0 \end{cases}$$

## Another dual problem

The flow problem can be put on dual form:

The vector $\bar{y}$ contains $|V|+|E|$ numbers. They are $g_i$ for each node $v_i$ and $\gamma_j$ for each edge $e_j$.

Minimize

$$\sum_j \gamma_j c_j$$

when

$$\begin{cases} g_i - g_j + \gamma_k \geq 0 & \text{om} \quad e_k = (v_i, v_j) \\ g_n - g_1 \geq 1, \quad \gamma_j \geq 0 & \text{for all} \quad j \end{cases}$$

The solution to this problem generates a minimal cut $(S, V-S)$ and an assignment of values $g_i = 0$ if $v_i \in S$, $g_i = 0$ otherwise. $\gamma_j = 1$ if $e_j$ goes from $S$ to $V-S$, $\gamma_j = 0$ otherwise.

## Other types of problems

Several of the problems we have seen are of integer character in the sense that the solutions have the form: Choose this object (1) and do not choose that object (0). But LP mostly gives solutions that are real numbers. We can define a new set of LP-problems called Integer Programming Problems by demanding that the solutions should have integer values.

Ex: Minimal spanning trees.

We can start with the LP-problem

Minimize

$$\sum_e x_e w(e)$$

when

$$\begin{cases} \sum_{e \in In(v)} x_e \geqslant 1 & \text{for all nodes } v \\ \sum_e x_e = |V| - 1 \end{cases}$$

This will almost give us a MST. But we must add the requirement that all $x_e$ should be 0 or 1. We can write this as

$x_e \in \{0, 1\}$ for all edges

This gives us an Integer Programming Problem.

**Subset Sum**

We can look at the special problem $A = \{2, 7, 9, 12, 16\}$ and $M = 23$ and ask $M$ can be written as a sum of elements from $A$.

We can then state the following IP-problem:

Maximize 1
when

$$\begin{cases} 2x_1 + 7x_2 + ... + 16x_5 = 23 \\ x_i \in \{0, 1\} \end{cases}$$

Even if it is nice to be able to state the problem as an IP-problem there is a catch: There is no known efficient algorithm for solving IP-problems!