Lecture 1: linear optimization: introduction

- Definition of cost / objective function
- Example of cost functions, affine functions, linear functions
- Definition of constraints
- Example of constraints, linear constraints
- Linear programs
- General form of a linear program
- Sigma notation
- Extended example 1: the transportation problem
- Google maps
- Extended example 2: the shortest path problem

What is optimization?

Minimize a cost or objective function (for ex. cost of production)

or

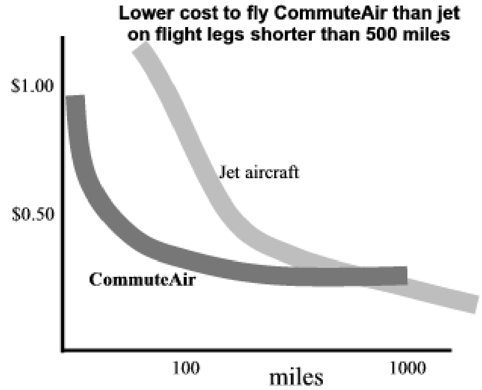
Maximize a cost or objective function (for ex. profit)

with respect to constraints

- Employee cannot work more than x hours a day
- Only three people can use the same machine at a time
- The pipeline's maximal fuel throughput is y

i.e. find a solution that is optimal within limits given

What is a cost function?

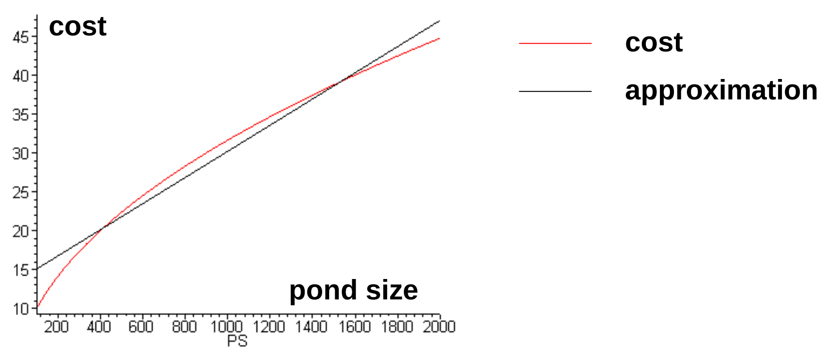


$$\text{cost} = f(\text{variables})$$

Example: cost of a mile as a function of the distance

[http://www.skyaid.org]

Linear or affine cost functions

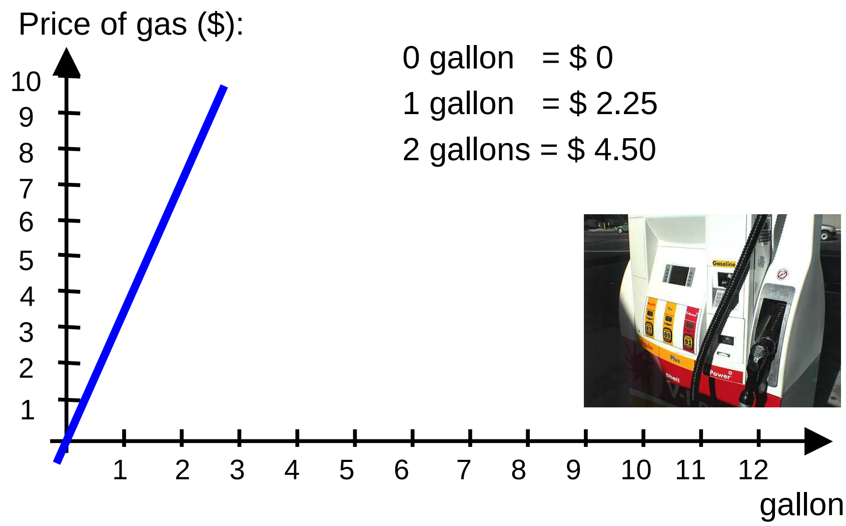


For some application, some cost functions look almost like "lines", i.e. are linear or affine. Example here: cost of building a dam as a function of the size of the pond

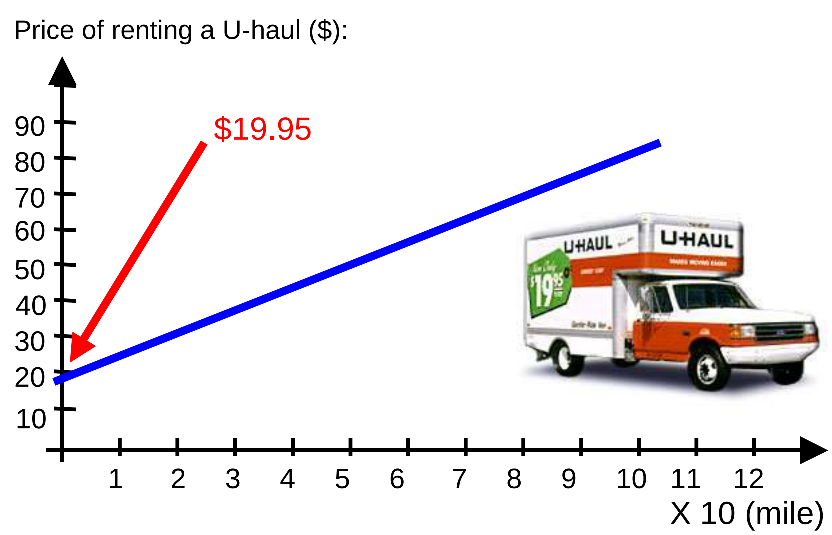
[http://home.hetnet.nl/~krekelberg/chapter3.htm]

Linar functions

0 gallon = $ 0
1 gallon = $ 2.25
2 gallons = $ 4.50



Affine functions



Linear or affine cost functions: formal definition

Minimizing the affine cost function

$$c(x_1, x_2) = 5 + 2x_1 + 3x_2$$

is the same as minimizing the linear cost function

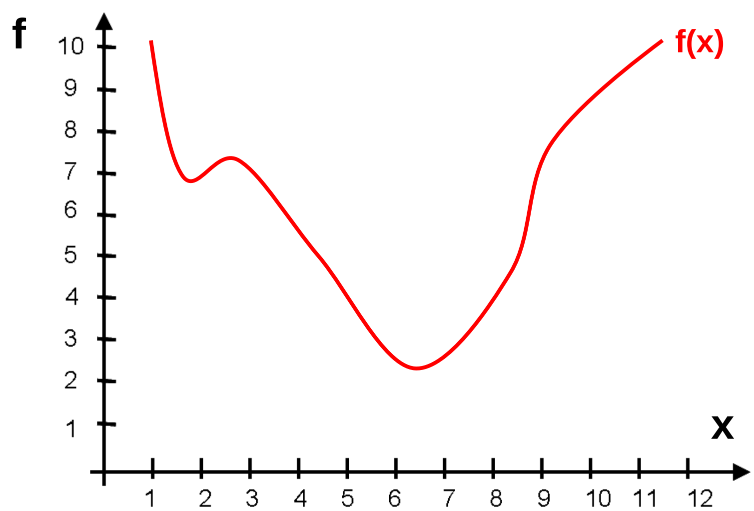
$$c(x_1, x_2) = 2x_1 + 3x_2$$

A more general expression of the cost function:

$$c(x_1, x_2, \cdots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n$$

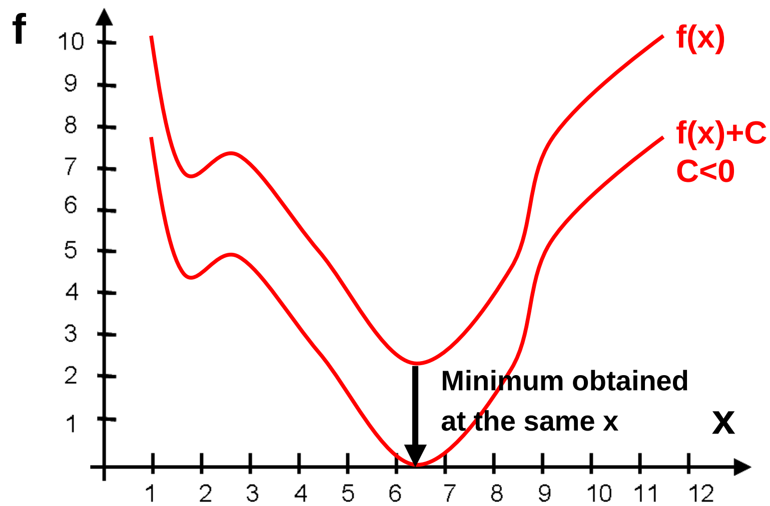
Minimizing affine or linear function is the same

Minimizing a function f(x)



Minimizing affine or linear function is the same

Minimizing a function f(x) or f(x)+c is the same



Example: cost of building a wall

Cost of a pound of cement ($ per lb)	a_1
Cost of a feet of steel beam ($ per ft)	a_2
Weight of cement (lb)	x_1
Length of steel beam (ft)	x_2

Total cost ($) $c(x_1, x_2) = a_1 x_1 + a_2 x_2$

Note that none of the variables above has the same unit!

Note however that $a_1 x_1$ **and** $a_2 x_2$ **and** $c(x_1, x_2)$ **have the same unit**

What is a constraint?

A constraint is a condition on variables which restricts the values they can take

Your maximal budget for cement is $c_{\max}$

$$a_1 x_1 \leq c_{\max}$$

Your minimal budget for steel is $s_{\min}$

$$a_2 x_2 \geq s_{min}$$

You want to spend twice as much for steel as for cement

$$a_2 x_2 \geq 2 a_1 x_1$$

You want to spend a given minimum amount for the wall $a_{\min}$

$$a_1 x_1 + a_2 x_2 \geq a_{\min}$$

Summary

Your optimization program incorporating all your constraints can be formulated as follows.

$$\begin{array}{ll} \textbf{Minimize:} & c(x_1, x_2) = a_1 x_1 + a_2 x_2 \\ \textbf{Subject to:} & a_1 x_1 \leq c_{max} \\ & a_2 x_2 \geq s_{min} \\ & a_1 x_1 + a_2 x_2 \geq a_{min} \\ & a_2 x_2 \geq 2 a_1 x_1 \end{array}$$

Constraints in the form of equalities (I)

Sometimes, constraints are given in the form of equalities

Example: you want to spend exactly twice as much for steel as for cement:

$$a_2 x_2 = 2 a_1 x_1$$

This is exactly the same as

$$a_2 x_2 \geq 2 a_1 x_1 \quad \textbf{and} \quad a_2 x_2 \leq 2 a_1 x_1$$

Constraints in the form of equalities (II)

So you could rewrite the program in the following form:

$$\begin{array}{ll} \textbf{Minimize:} & c(x_1, x_2) = a_1 x_1 + a_2 x_2 \\ \textbf{Subject to:} & a_1 x_1 \leq c_{max} \\ & a_2 x_2 \geq s_{min} \\ & a_1 x_1 + a_2 x_2 \geq a_{min} \\ & a_2 x_2 \geq 2 a_1 x_1 \\ & a_2 x_2 \leq 2 a_1 x_1 \end{array}$$

One can thus assume that all constraints are always given in the form of inequalities.

General form for a linear program

So you could rewrite the program in the following form:

$$
\begin{array}{lll}
\textbf{min:} & c_1x_1 + c_2x_2 + \cdots + c_Nx_N & \\
\textbf{s.t.:} & a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,j}x_j \cdots + a_{1,N}x_N & \leq b_1 \\
 & a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,j}x_j \cdots + a_{2,N}x_N & \leq b_2 \\
 & \vdots & \vdots \\
 & a_{M,1}x_1 + a_{M,2}x_2 + \cdots + a_{M,j}x_j \cdots + a_{M,N}x_N & \leq b_M
\end{array}
$$

Sigma notation

So you could rewrite the program in the following form:

$$
\begin{array}{lll}
\textbf{min:} & \sum_{j=1}^{N} c_jx_j & \\
\textbf{s.t.:} & \sum_{j=1}^{N} a_{1,j}x_j & \leq b_1 \\
 & \sum_{j=1}^{N} a_{2,j}x_j & \leq b_2 \\
 & \vdots & \vdots \\
 & \sum_{j=1}^{N} a_{M,j}x_j & \leq b_M
\end{array}
$$

Example: the transportation problem (I)

Paul's farm produces 4 tons of apples per day $s_p = 4$
Ron's farm produces 2 tons of apples per day $s_r = 2$
Max's factory needs 1 ton of apples per day $d_m = 1$
Bob's factory needs 5 tons of apples per day $d_b = 5$

George owns both farms and factories. He is paying the cost of shipping all the apples from the farms to the factories.

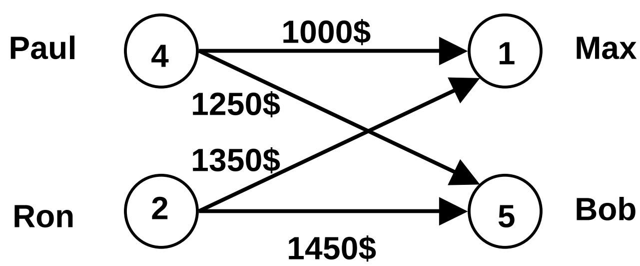
The shipping costs for George are:

- **Paul →Max: 1000$ per ton** $c_{pm} = 1000$ x_{pm}
- **Ron →Max: 1350$ per ton** $c_{rm} = 1350$ x_{rm}
- **Paul →Bob: 1250$ per ton** $c_{pb} = 1250$ x_{pb}
- **Ron →Bob: 1450$ per ton** $c_{rb} = 1450$ x_{rb}

What is the best way to ship the apples?

Example: the transportation problem (II)

George pays for the shipping

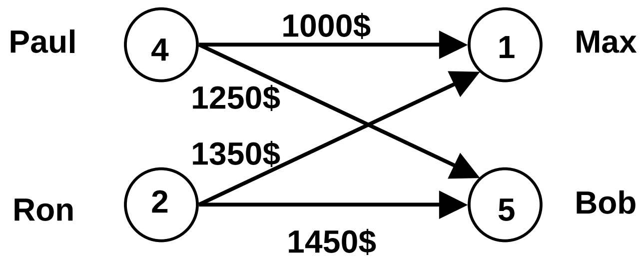


Example: the transportation problem (III)

min: $1000x_{pm} + 1350x_{rm} + 1250x_{pb} + 1450x_{rb}$

Subject to:

$$x_{pm} + x_{rm} = 1$$
$$x_{pb} + x_{rb} = 5$$
$$x_{pm} + x_{pb} = 4$$
$$x_{rm} + x_{rb} = 2$$
$$x_{pm} \geq 0,\ x_{rm} \geq 0,\ x_{pb} \geq 0,\ x_{rb} \geq 0$$

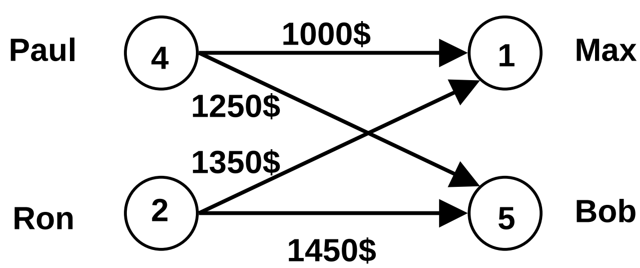


Example: the transportation problem (IV)

min: $x_{pm}c_{pm} + x_{rm}c_{rm} + x_{pb}c_{pb} + x_{rb}c_{rb}$

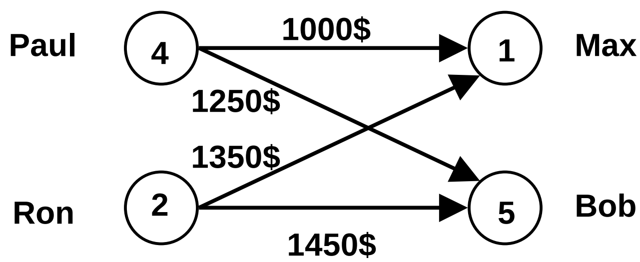
Subject to:

$$x_{pm} + x_{rm} = d_m$$
$$x_{pb} + x_{rb} = d_b$$
$$x_{pm} + x_{pb} = s_p$$
$$x_{rm} + x_{rb} = s_r$$
$$x_{pm} \geq 0\ x_{rm} \geq 0,\ x_{pb} \geq 0,\ x_{rb} \geq 0$$



General form of the transportation problem

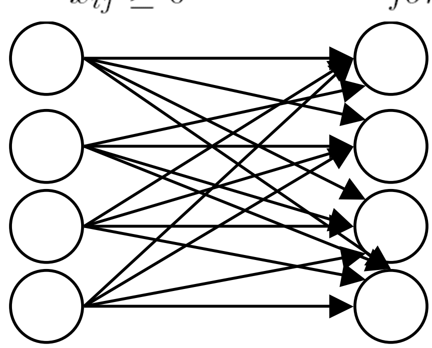
$$
\begin{array}{lll}
\textbf{min:} & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} & \\
\textbf{Subject to:} & \sum_{i=1}^{m} x_{ij} = d_j & j = 1,\cdots,n \\
& \sum_{j=1}^{n} x_{ij} = s_i & i = 1,\cdots,m \\
& x_{ij} \geq 0 & for\ all\ i,j
\end{array}
$$



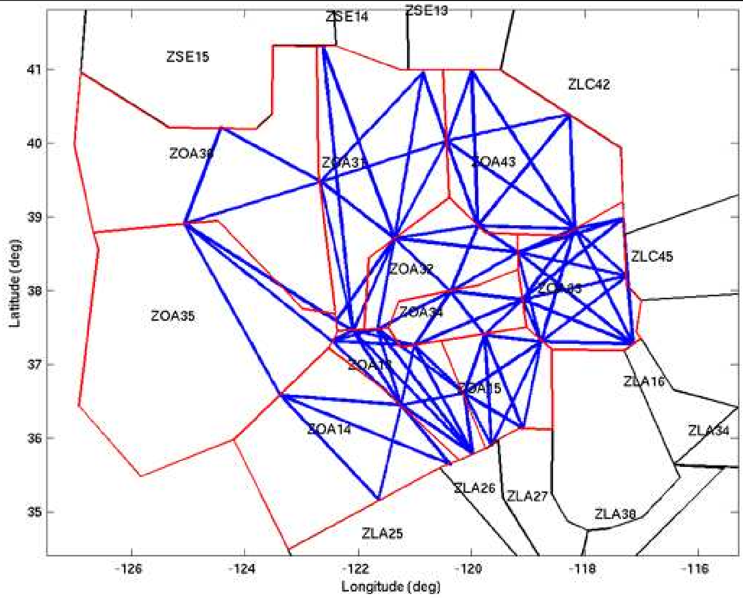
Please, be lazy, do not write pages of equations…

Use summations, they leave you more time to go to the movies

$$
\begin{array}{lll}
\textbf{min:} & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} & \\
\textbf{Subject to:} & \sum_{i=1}^{m} x_{ij} = d_j & j = 1,\cdots,n \\
& \sum_{j=1}^{n} x_{ij} = s_i & i = 1,\cdots,m \\
& x_{ij} \geq 0 & for\ all\ i,j
\end{array}
$$

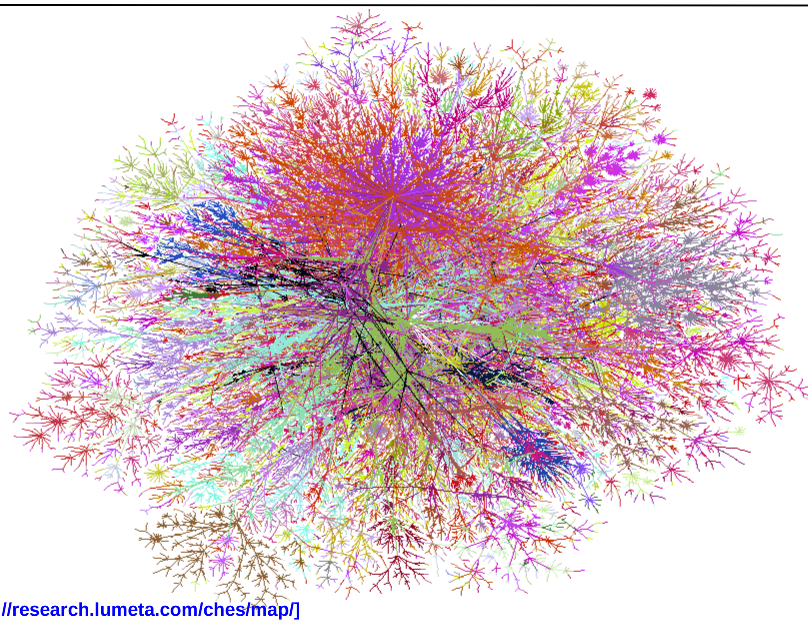


Example: a « small » network (air traffic control)



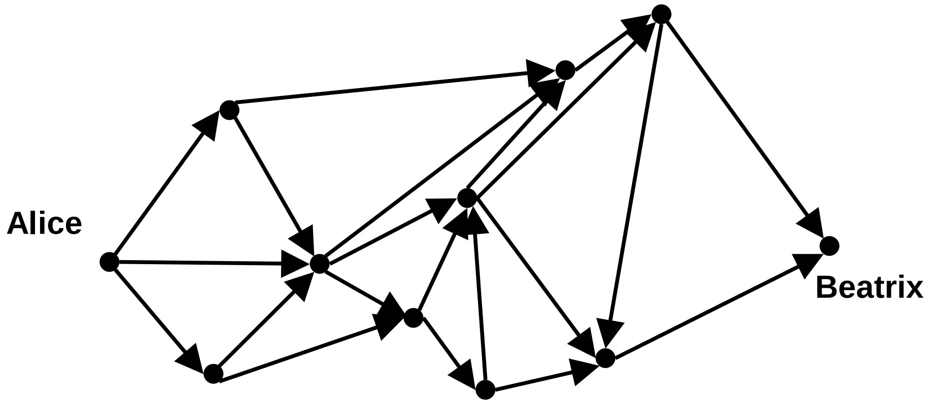
[Robelin, Sun, Bayen, tech. rep., 2005]

Example: a « large » network (the internet)

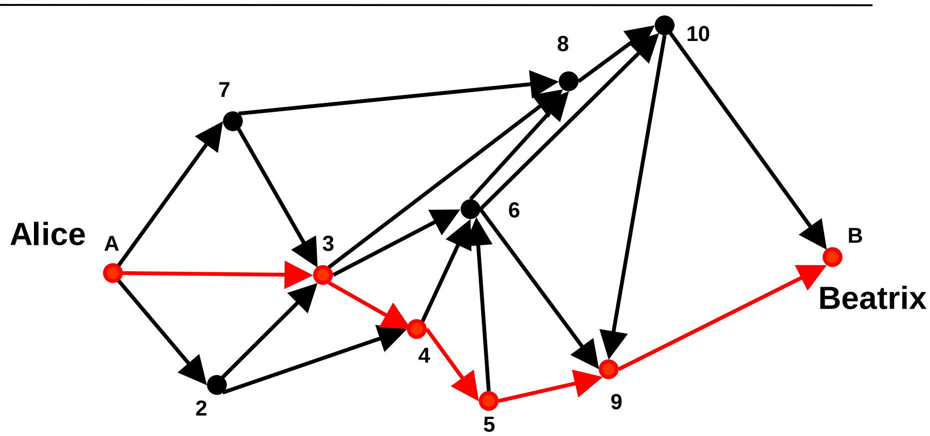


[http://research.lumeta.com/ches/map/]

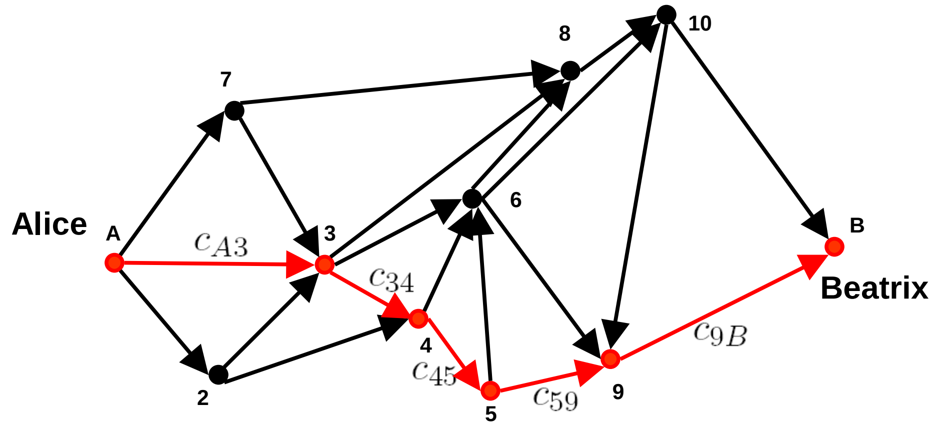
Example: shortest path



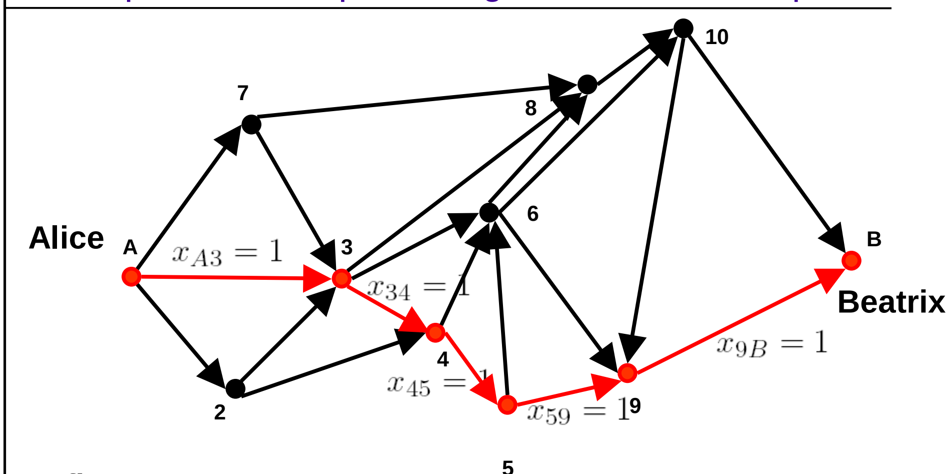
Example: shortest path



Example: shortest path: length of the shortest path



Example: shortest path: length of the shortest path

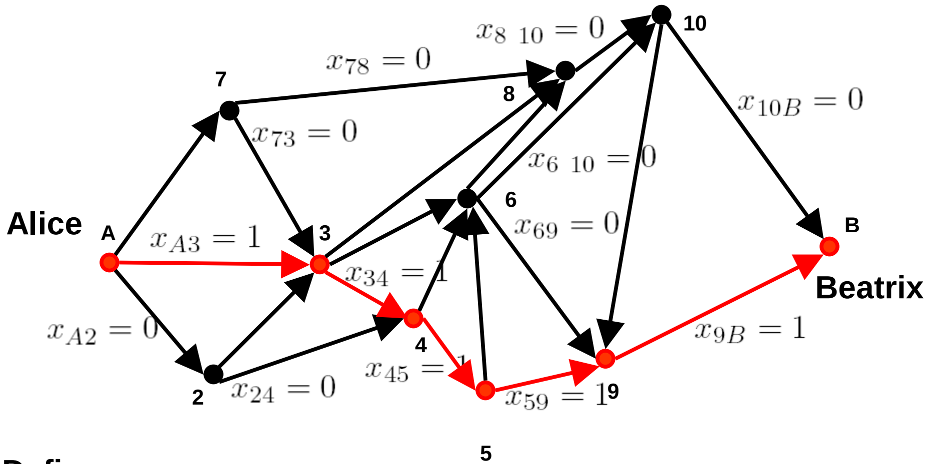


Define

$x_{ij} = 1$ **For every (i,j) on the shortest path**

$x_{ij} = 0$ **For every (i,j) not on the shortest path**

Example: shortest path: length of the shortest path

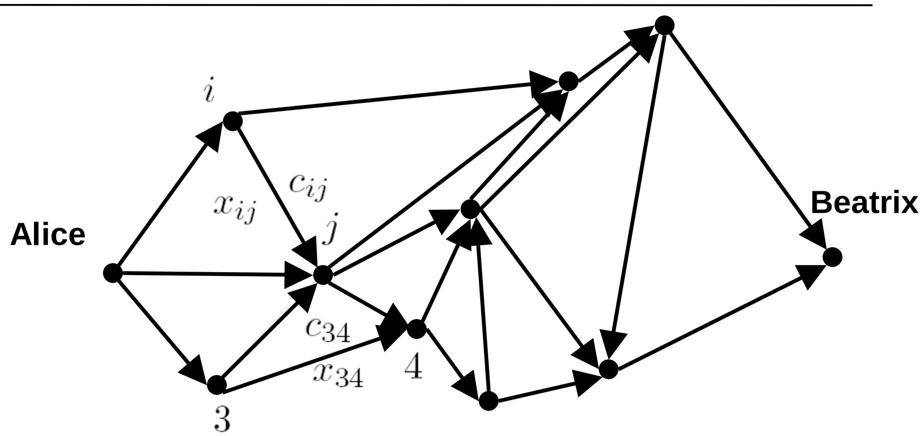


Define

$x_{ij} = 1$ **For every (i,j) on the shortest path**

$x_{ij} = 0$ **For every (i,j) not on the shortest path**

Example: shortest path: choice of path



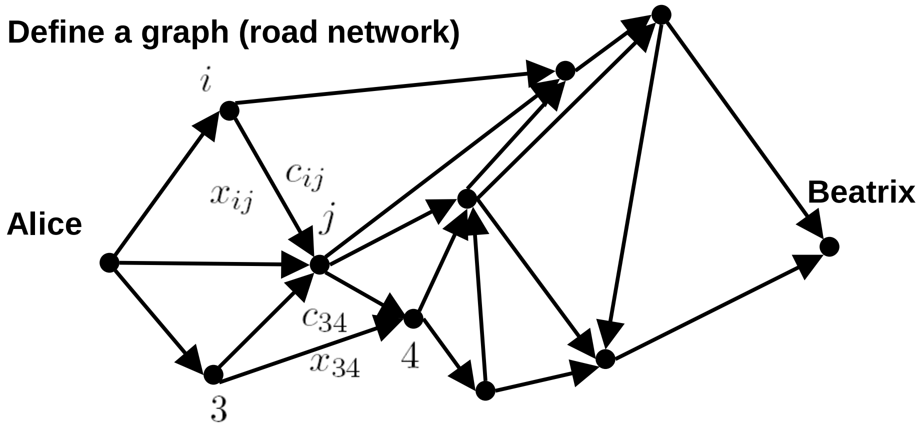
Define

$x_{ij} = 1$ **For every (i,j) on the shortest path**

$x_{ij} = 0$ **For every (i,j) not on the shortest path**

Example: shortest path

Define a graph (road network)

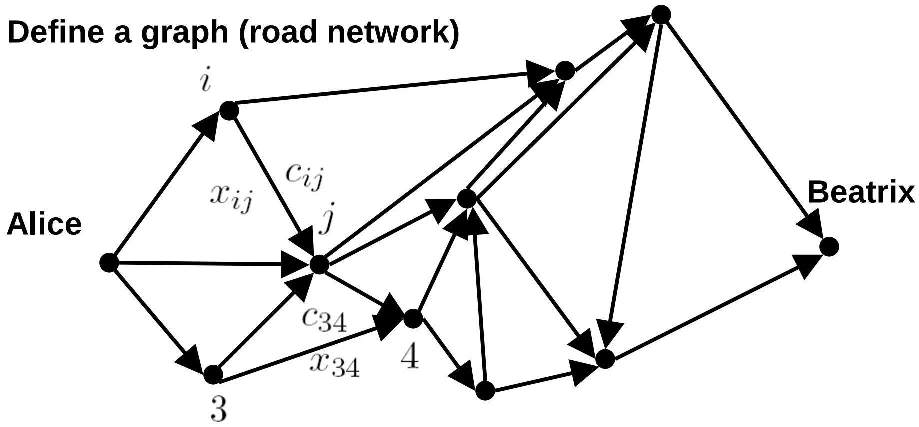


Call c_{ij} the cost to go from i to j (for exmple fuel burned)

For example c_{34} is the cost to go from node 3 to node 4

Example: shortest path

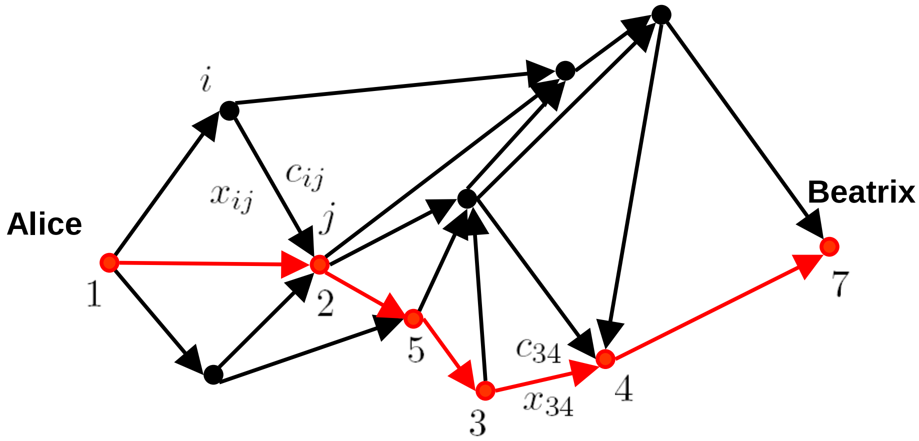
Define a graph (road network)



Take $x_{ij} = 1$ if Alice decides to go through link (i,j), zero otherwise

For example $x_{34} = 1$ if Alice decides to use route (3,4)

Example: shortest path

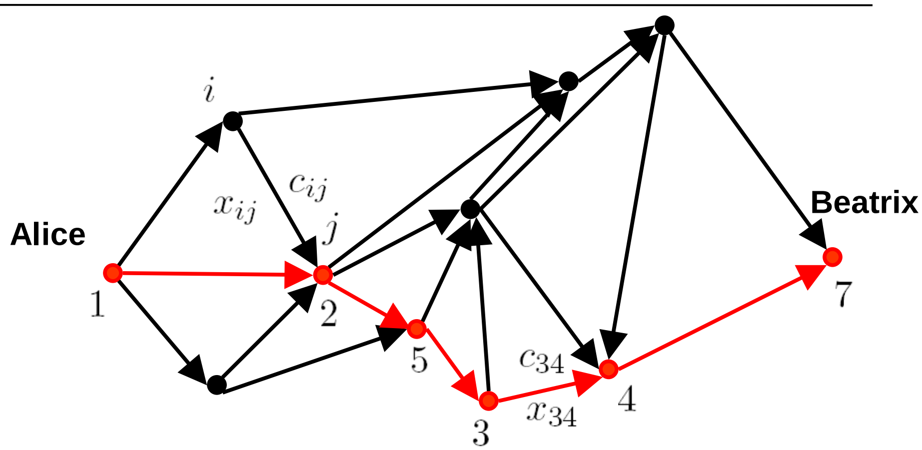


$x_{12} = x_{25} = x_{53} = x_{34} = x_{47} = 1$

All other x_{ij} **are zero**

Total length of this path: $c_{12} + c_{25} + c_{53} + c_{34} + c_{47}$

Example: shortest path



Total length:

$$\sum_{(i,j) \text{ chosen on path}} c_{ij} = \sum_{(i,j) \text{ chosen on path}} x_{ij} c_{ij} = \sum_{\text{all } (i,j)} x_{ij} c_{ij}$$

Example: shortest path

Minimize: $Z = \sum_{j \in N_A} c_{Aj} x_{Aj} + \sum_{i=1}^{10} \sum_{j \in N_i} c_{ij} x_{ij} + \sum_{j \in N_B} c_{jB} x_{jB}$

Total length

Cost of arc (i,j)

1 if link (i,j) chosen
0 otherwise

N_i **set of nodes j with direct connections to node i**

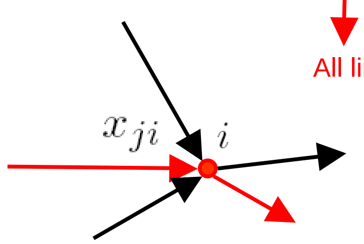
Example: shortest path

minimize: $Z = \sum_{j \in N_A} c_{Aj} x_{Aj} + \sum_{i=1}^{10} \sum_{j \in N_i} c_{ij} x_{ij} + \sum_{j \in N_B} c_{jB} x_{jB}$

such that: $\sum_{j \in N_i} x_{ji} = \sum_{j \in N_i} x_{ij}, \quad i = 1, \cdots, 10$

All links leaving node i

All links arriving at node i



N_i **set of nodes j with direct connections to node i**

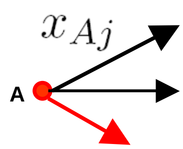
Example: shortest path

minimize: $$Z = \sum_{j \in N_A} c_{Aj} x_{Aj} + \sum_{i=1}^{10} \sum_{j \in N_i} c_{ij} x_{ij} + \sum_{j \in N_B} c_{jB} x_{jB}$$

such that: $$\sum_{j \in N_i} x_{ji} = \sum_{j \in N_i} x_{ij}, \quad i = 1, \cdots, 10$$

$$\sum_{j \in N_A} x_{Aj} = 1$$

Starting from A, Alice can only take one path



N_A **set of nodes j with direct connections to node A**

Example: shortest path

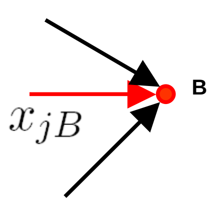
minimize: $$Z = \sum_{j \in N_A} c_{Aj} x_{Aj} + \sum_{i=1}^{10} \sum_{j \in N_i} c_{ij} x_{ij} + \sum_{j \in N_B} c_{jB} x_{jB}$$

such that: $$\sum_{j \in N_i} x_{ji} = \sum_{j \in N_i} x_{ij}, \quad i = 1, \cdots, 10$$

$$\sum_{j \in N_A} x_{Aj} = 1$$

$$\sum_{j \in N_B} x_{jB} = 1$$

Arriving at B, one can only take one path



N_B **set of nodes j with direct connections to node B**

Example: shortest path

minimize: $$Z = \sum_{j \in N_A} c_{Aj} x_{Aj} + \sum_{i=1}^{10} \sum_{j \in N_i} c_{ij} x_{ij} + \sum_{j \in N_B} c_{jB} x_{jB}$$

such that: $$\sum_{j \in N_i} x_{ji} = \sum_{j \in N_i} x_{ij}, \quad i = 1, \cdots, 10$$

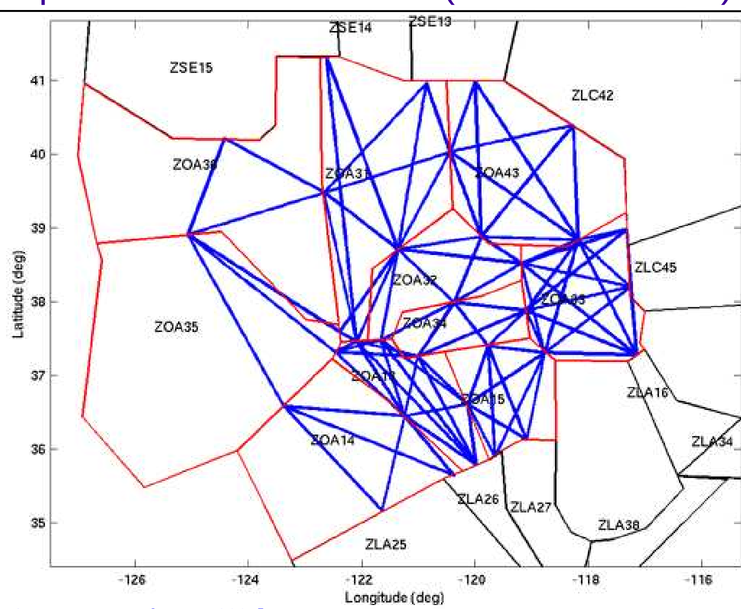
$$\sum_{j \in N_A} x_{Aj} = 1$$

$$\sum_{j \in N_B} x_{jB} = 1$$

$$x_{ij} \geq 0, \; x_{jB} \geq 0, \quad x_{Aj} \geq 0$$

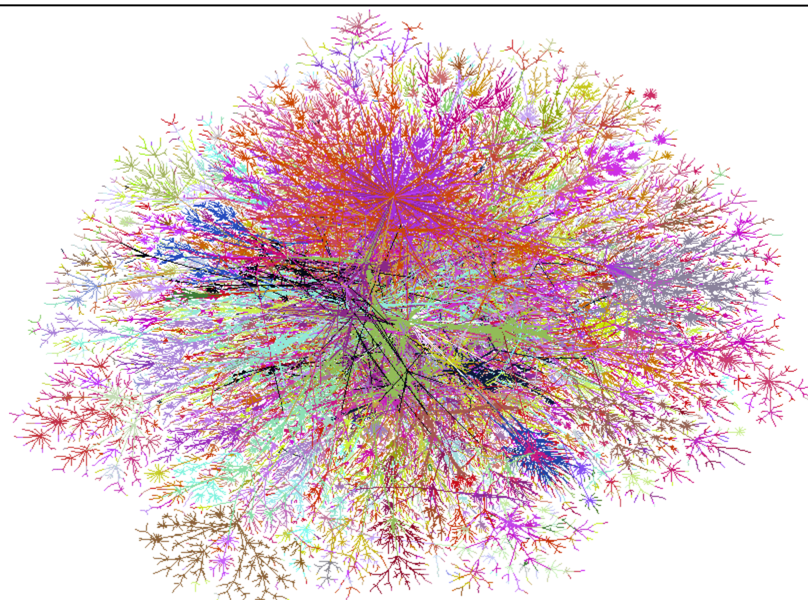
N_i **set of nodes j with direct connections to node i**

Example: a « small » network (air traffic control)



[Robelin, Sun, Bayen, tech. rep., 2005]

Example: a « large » network (the internet)



[http://research.lumeta.com/ches/map/]