CIND-221: Taller 2

Felipe Osorio

f.osoriosalgado@uandresbello.edu

Facultad de Ingeniería, UNAB

Taller 2:

Ejercicio:

El precio de una máquina nueva es de 500. El costo de mantenimiento es 200 el 1er año, 400 el 2do año, 600 el 3er año y 800 el 4to año de uso. Suponiendo que la máquina no tiene valor de reventa.

Halle el costo mínimo de comprar y utilizar la máquina durante un lapso de 4 años, si se compra una máquina al comienzo del primer año.

(a) Dibuje la red $G = (N, A)$, con $N = \{1, 2, 3, 4, 5\}$ y obtenga los costos de los arcos c_{ij}.

(b) Escriba el problema en AMPL, obtenga la ruta óptima (para ir del nodo 1 al 5).

Usando AMPL

Archivo **maquinas.dat**

```
set KNOTS := 1 2 3 4 5;

param entr := 1;
param exit := 5;

param: ROADS: cost:=
1 2 700,
1 3 1100,
1 4 1700,
1 5 2500,
2 3 700,
2 4 1100,
2 5 1700,
3 4 700,
3 5 1100,
4 5 700;

```

Archivo **rutas.mod**

```
set KNOTS; # nodos

param entr symbolic in KNOTS;             # Entranda a la red
param exit symbolic in KNOTS, <> entr;   # Salida desde la red

set ROADS within (KNOTS diff {exit}) cross (KNOTS diff {entr});

param cost {ROADS} >=0;      # costos
var x {(i,j) in ROADS} >=0; # 1 ssi (i,j) en la ruta más corta

minimize total: sum{(i,j) in ROADS} cost[i,j] * x[i,j];

subject to start: sum {(entr,j) in ROADS} x[entr,j] = 1;

subject to balance {k in KNOTS diff{entr,exit}}:
  sum {(i,k) in ROADS} x[i,k] = sum {(k,j) in ROADS} x[k,j];

```

Usando AMPL

Ejecutando la optimización en la consola de AMPL:

```
ampl: reset;
ampl: model rutas.mod;
ampl: data maquinas.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 22.1.2: optimal solution; objective 2200
0 simplex iterations

```

Salida:

```
ampl: display total;
total = 2200

ampl: display x;
x :=
1 2    0
1 3    1
1 4    0
1 5    0
2 3    0
2 4    0
2 5    0
3 4    0
3 5    1
4 5    0
;

```

Usando AMPL

En la consola también podemos ejecutar el archivo **maquinas.run**:

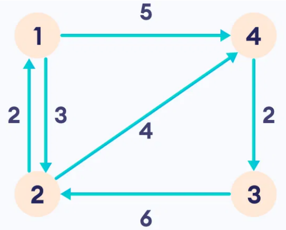
```
ampl: include maquinas.run;
CPLEX 22.1.2: optimal solution; objective 2200
0 simplex iterations
total = 2200

x :=
1 2    0
1 3    1
1 4    0
1 5    0
2 3    0
2 4    0
2 5    0
3 4    0
3 5    1
4 5    0
;

```

Ejercicio:

Considere la red $G = (N, A)$, con $N = \{1, 2, 3, 4\}$ dada por:



Describiremos cómo aplicar el algoritmo de Floyd en AMPL.

Usando AMPL

Archivo **network.dat**

```
set NODES := 1 2 3 4;

param cost:
   1 2 3 4 :=
1  . 3 . 5
2  2 . . 4
3  . 1 . .
4  . . 2 . ;
```

Archivo **floyd.mod**

```
set NODES;

# Representa los pesos iniciales
param cost {NODES, NODES} default Infinity;

# Distancias del camino más corto
param d {NODES, NODES};

# Nodos predecesores para la reconstrucción de rutas
param pred {NODES, NODES};

```

Archivo **floyd.run**

```
model floyd.mod;
data network.dat;

# Inicializar 'd' y 'pred'
for {i in NODES, j in NODES} {
    if i = j then {
        let d[i,j] := 0;
        let pred[i,j] := 0; # o un valor distinto
    } else if cost[i,j] < Infinity then {
        let d[i,j] := cost[i,j];
        let pred[i,j] := i;
    } else {
        let d[i,j] := Infinity;
        let pred[i,j] := 0; # o un valor distinto
    }
}

```

Archivo **floyd.run** (...continuación)

```
# iteraciones de Floyd-Warshall
for {k in NODES} {
    for {i in NODES} {
        for {j in NODES} {
            if d[i,k] + d[k,j] < d[i,j] then {
                let d[i,j] := d[i,k] + d[k,j];
                let pred[i,j] := pred[k,j];
            }
        }
    }
}

display d;
display pred;
```

Usando AMPL

Ejecutando el archivo **floyd.run**:

```
ampl: reset;
ampl: include floyd.run;
d :=
1 1    0
1 2    3
1 3    7
1 4    5
2 1    2
2 2    0
2 3    6
2 4    4
3 1    3
3 2    1
3 3    0
3 4    5
4 1    5
4 2    3
4 3    2
4 4    0
;

```

Usando AMPL

Ejecutando el archivo **floyd.run** (...continuación)

```
pred :=
1 1   0
1 2   1
1 3   4
1 4   1
2 1   2
2 2   0
2 3   4
2 4   2
3 1   2
3 2   3
3 3   0
3 4   2
4 1   2
4 2   3
4 3   4
4 4   0
;

```

Es decir,

$$\mathtt{d} = \begin{pmatrix} 0 & 3 & 7 & 5 \\ 2 & 0 & 6 & 4 \\ 3 & 1 & 0 & 5 \\ 5 & 3 & 2 & 0 \end{pmatrix}, \qquad \mathtt{pred} = \begin{pmatrix} 0 & 1 & 4 & 1 \\ 2 & 0 & 4 & 2 \\ 2 & 3 & 0 & 2 \\ 2 & 3 & 4 & 0 \end{pmatrix}.$$

corresponden a las matrices $\boldsymbol{C}_4$ y $\boldsymbol{S}_4$, respectivamente.