CIND-221: Taller 3

Felipe Osorio

f.osoriosalgado@uandresbello.edu

Facultad de Ingeniería, UNAB

Árbol de expansión mínima:

Propiedad:

Todo árbol de expansión tiene exactamente $n-1$ arcos.

Definición:

Considere una red $G=(N,A)$ con $N=\{1,\ldots,n\}$, y sea

$$x_{ij}=\begin{cases}1, & \text{si el arco } (i,j) \text{ está en el árbol,}\\ 0, & \text{en otro caso.}\end{cases}$$

Defina $\boldsymbol{X}=(x_{ij})$ para $(i,j)\in A$.

Idea:

El árbol de expansión mínima tiene por objetivo hallar la subred $T^*=(N,A^*)$ con $A^*=\{(i,j)\in A : x_{ij}=1\}$ corresponde a los arcos seleccionados del árbol de expansión.

Árbol de expansión mínima:

Formulación como un problema de PL:

$$\min_{X} \sum_{(i,j)\in A} \phi_{ij} x_{ij},$$

sujeto a:

$$\sum_{(i,j)\in A} x_{ij} = n-1,$$

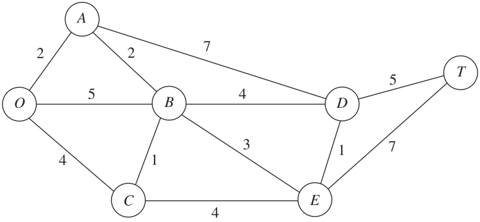
$$\sum_{(i,j)\in \delta(S)} x_{ij} \geq 1, \quad \forall S \subset N, S \neq N, S \neq \varnothing$$

$$\sum_{(i,j)\in A(S)} x_{ij} \leq |S| - 1, \quad \forall S \subset N, S \neq N, S \neq \varnothing,$$

donde $\delta(S) \subset A$ es un subconjunto de arcos con un extremo en S y el otro en $N - S$, mientras que $A(S) \subset A$ es un subconjunto de arcos cuyos extremos están en el conjunto $S \in N$.

Ejercicio:

La administración de una ciudad debe determinar los caminos en que se debe tender líneas telefónicas para conectar ciertas estaciones para usar una longitud total mínima de cable. Considere la siguiente red:



Usando AMPL

Archivo **seervada.dat**

```
set Nodos := 1 2 3 4 5 6 7
;

param :P: c:
1 2 3 4 5 6 7 :=
1 . 2 5 4 . . .
2 . . 2 . 7 . .
3 . . . 1 4 3 .
4 . . . . . 4 .
5 . . . . . 1 5
6 . . . . . . 7
7 . . . . . . .
;

```

Usando AMPL

Archivo **arbol.mod**

```
set Nodos ordered;             # se asume orden ascendente
set P within {Nodos,Nodos}; # el producto cruz D x D,
                            # se redefine con el parametro c
param c {P};                # matriz de costos

# el conjunto de todos los posibles arcos que unen a cada nodo
set Conex {i in Nodos} within P := {(u,v) in P: u==i or v==i};
# cardinalidad del conjunto D
param n := card(Nodos);
# numero de elementos en el conjunto potencia
set S := 0 .. (2**n-1);
# conjunto potencia de los nodos
set CPOT {k in S}:={i in Nodos: (k div 2**(ord(i,Nodos)-1)) mod 2
     ==1};

var x {P} binary; # variable de decisión

minimize costos: sum{(i,j) in P} c[i,j] * x[i,j]; # función objetivo

```

Usando AMPL

Archivo **arbol.mod** (...continuación)

```
# aseguramos que todos los nodos deben tener arcos
subject to arcos {i in Nodos}: sum{(u,v) in Conex[i]} x[u,v] >= 1;

# nuestra solucion tiene que ser un arbol
# si G tiene |V| vertices, entonces T tiene |V|-1 aristas:
subject to tamano_del_arbol: sum{(i,j) in P} x[i,j] = n -1;

# esta restricción indica que no hay ciclos entre nodos
subject to sin_ciclos{k in S: card({i in CPOT[k],j in CPOT[k]:(i,j)
     in P})>=3}:
sum{i in CPOT[k], j in CPOT[k]: (i,j) in P} x[i,j] <= card(CPOT[k])
     - 1;

```

Usando AMPL

Ejecutando la optimización en la consola de AMPL:

```
ampl: reset;
ampl: model arbol.mod;
ampl: data seervada.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 22.1.2: optimal solution; objective 14
1 simplex iterations

```

Usando AMPL

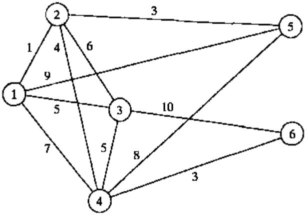
Salida:

```
ampl: display x;
x :=
1 2   1
1 3   0
1 4   0
2 3   1
2 5   0
3 4   1
3 5   0
3 6   1
4 6   0
5 6   1
5 7   1
6 7   0
;

```

Ejercicio:

Considere la siguiente red $G = (N, A)$, donde $N = \{1, 2, 3, 4, 5, 6\}$, con:



Archivo **sesion7.dat**

```
set Nodos := 1 2 3 4 5 6;

param :P: c:
   1 2 3 4 5 6 :=
1 . 1 5 7 9 .
2 . . 6 4 3 .
3 . . . 5 . 10
4 . . . . 8 3
5 . . . . . .
6 . . . . . .
;

```

Usando AMPL

Ejecutando la optimización en la consola de AMPL:

```
ampl: reset;
ampl: model arbol.mod;
ampl: data sesion7.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 22.1.2: optimal solution; objective 16
0 simplex iterations

```

Salida:

```
ampl: display x;
x :=
1 2    1
1 3    0
1 4    0
1 5    0
2 3    0
2 4    1
2 5    1
3 4    1
3 6    0
4 5    0
4 6    1
;

```

Usando AMPL

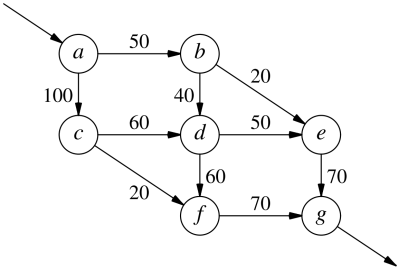
En la consola también podemos ejecutar el archivo **arbol.run**:

```
ampl: include arbol.run;
CPLEX 22.1.2: optimal solution; objective 16
0 simplex iterations
x :=
1 2   1
1 3   0
1 4   0
1 5   0
2 3   0
2 4   1
2 5   1
3 4   1
3 6   0
4 5   0
4 6   1
;

```

Ejercicio:

Considere la siguiente red $G = (N, A)$, donde $N = \{a, b, c, d, e, f, g\}$, con nodo fuente $\boldsymbol{a}$ y nodo sumidero $\boldsymbol{g}$



Se desea determinar el flujo máximo de tráfico en esta red vehicular.

Archivo **trafico.dat**

```
set INTER := a b c d e f g ;

param entr := a ;
param exit := g ;

param:  ROADS:  cap :=
          a b     50,  a c    100
          b d     40,  b e     20
          c d     60,  c f     20
          d e     50,  d f     60
          e g     70,  f g     70;
```

Usando AMPL

Archivo **flujo.mod**

```
set INTER;  # intersecciones

param entr symbolic in INTER;            # entrada a la red
param exit symbolic in INTER, <> entr;  # salida de la red

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

param cap {ROADS} >= 0;                            # capacidades
var Traff {(i,j) in ROADS} >= 0, <= cap[i,j];  # cargas de tráfico

maximize Entering_Traff: sum {(entr,j) in ROADS} Traff[entr,j];

subject to Balance {k in INTER diff {entr,exit}}:
  sum {(i,k) in ROADS} Traff[i,k] = sum {(k,j) in ROADS} Traff[k,j];

```

Usando AMPL

Ejecutando la optimización en la consola de AMPL:

```
ampl: reset;
ampl: model flujo.mod;
ampl: data trafico.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 22.1.2: optimal solution; objective 130
2 simplex iterations

```

Salida:

```
ampl: display Traff;
Traff :=
a b   50
a c   80
b d   30
b e   20
c d   60
c f   20
d e   40
d f   50
e g   60
f g   70
;

```

Usando AMPL

Archivo **flujo.run**

```
reset;
model flujo.mod;
data trafico.dat;
option solver cplex;
solve;
display Traff;
```

Usando AMPL

Ejecutando el archivo **flujo.run**:

```
ampl: include flujo.run;
CPLEX 22.1.2: optimal solution; objective 130
2 simplex iterations
Traff :=
a b   50
a c   80
b d   30
b e   20
c d   60
c f   20
d e   40
d f   50
e g   60
f g   70
;

```

Ejercicio propuesto:

Obtener el flujo máximo en la siguiente red $G = (N, A)$, donde

$$N = \{f, a, b, c, d, s\},$$

con nodo fuente $\boldsymbol{f}$ y nodo sumidero $\boldsymbol{s}$