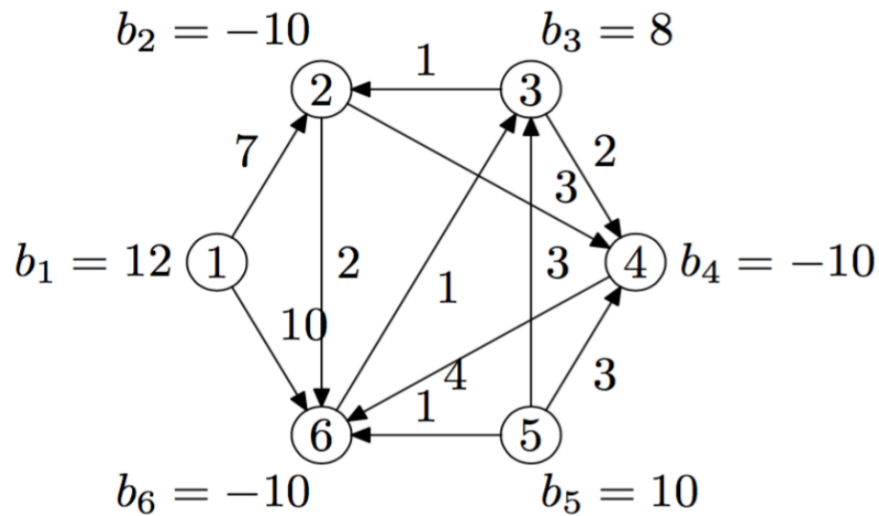


Consegna Laboratorio 7

PEPE SVEVA - 1743997
MEDAGLIA CLAUDIA- 1758095
POGGI MATTIA- 1762074

Descrizione Problema:



- 1 Formula matematicamente il problema di flusso di costo minimo con un modello di PLI

Le variabili decisionali:

$$x_{ij} \quad i = 1, \dots, 6 \quad j = 1, \dots, 6$$

La quale identifica gli archi, dove l'indice i identifica nodo uscente e l'indice j il nodo entrante

$$\begin{cases} \min(7x_{12} + 10x_{16} + 3x_{24} + 2x_{26} + x_{32} + 2x_{34} + \\ \quad 4x_{46} + 3x_{53} + 3x_{54} + x_{56} + x_{63}) \\ x_{12} + x_{16} & = 12 \\ x_{24} + x_{26} - x_{12} - x_{23} & = -10 \\ x_{32} + x_{34} - x_{63} - x_{53} & = 8 \\ x_{46} - x_{24} - x_{34} - x_{54} & = -10 \\ x_{53} + x_{54} + x_{56} & = 10 \end{cases}$$

Posso eliminare l'ultimo vincolo perché:

Ogni vincolo del problema di flusso a costo minimo può essere ottenuto attraverso la somma di tutti gli altri.

Quindi: uno (ed un solo) vincolo del problema (non importa quale) può essere eliminato in quanto ridondante, per convenzione abbiamo eliminato l'ultimo.

2 Implementa il modello in Opl scrivendo il file .mod ed il file .dat, inserendo una funzione main che generi il modello, lo risolva ed esegua la funzione di postprocessing per la scrittura della soluzione ottima su file

lab7-1.dat :

```

0  B=[12,-10,8,-10,10,-10];
   file=1;
2  Arcs = {
   // from, to, costo
4  < 1, 2, 7 >,
   < 1, 6, 10 >,
6  < 2, 4, 3 >,
   < 2, 6, 2 >,
8  < 3, 2, 1 >,
   < 3, 4, 2 >,
10 < 4, 6, 4 >,
   < 5, 3, 3 >,
12 < 5, 4, 3 >,
   < 5, 6, 1 >,
14 < 6, 3, 1 >
   };
16

```

lab7.mod

```
0  range Nodes = 1..6;
   int file=...;
2  int B[Nodes]=...;

4  tuple arc {
   key int fromnode;
6   key int tonode;
   float cost;
8  }

10 {arc} Arcs = ...;

12 dvar int+ x[a in Arcs];

14 minimize
   sum (a in Arcs)
16   x[a]*a.cost;

18 subject to{
   forall (i in Nodes)
20   ct:
       sum (<i,j,c> in Arcs) x[<i,j,c>]
22   - sum (<j,i,c> in Arcs) x[<j,i,c>] == B[i];
   }

24 execute{
26   var ofile=new IloOplOutputFile("lab7"+"-"+file+".txt");
   ofile.writeln("Objective =", cplex.getObjValue());
28   for (var a in Arcs)
       ofile.writeln("x", a.fromnode, a.tonode, "=", x[a]);
30   ofile.close();
   }

32 main{
   thisOplModel.generate();
34   if(cplex.solve()){
       thisOplModel.postProcess();
36   var obj = cplex.getObjValue();
       writeln("OBJ = ", cplex.getObjValue());
38   }
   }

40 }
```

La soluzione della funzione obiettivo: 116

il valore della soluzione ottima

$$\begin{cases} x_{12} = 12 \\ x_{16} = 0 \\ x_{24} = 2 \\ x_{26} = 0 \\ x_{32} = 0 \\ x_{34} = 8 \\ x_{46} = 0 \\ x_{53} = 0 \\ x_{54} = 0 \\ x_{56} = 10 \\ x_{63} = 0 \end{cases}$$

Quindi $\tilde{x}=[12 \ 0 \ 2 \ 0 \ 0 \ 8 \ 0 \ 0 \ 0 \ 10 \ 0]$

3 Supponi che la domanda del nodo 2 sia 12 e che la disponibilità del nodo 5 sia 12. Scrivi un nuovo file .dat e modifica la funzione main in modo che le due istanze del problema vengano risolte automaticamente in modo iterativo.

lab7-2.dat :

```
0  B=[12,-12,8,-10,12,-10];
   file=2;
2  Arcs = {
   // from, to, costo
4  < 1, 2, 7 >,
   < 1, 6, 10 >,
6  < 2, 4, 3 >,
   < 2, 6, 2 >,
8  < 3, 2, 1 >,
   < 3, 4, 2 >,
10 < 4, 6, 4 >,
   < 5, 3, 3 >,
12 < 5, 4, 3 >,
   < 5, 6, 1 >,
14 < 6, 3, 1 >
   };
16
```

lab7-1.dat

```
0  B=[12,-10,8,-10,10,-10];
   file=1;
2  Arcs = {
   // from, to, costo
4  < 1, 2, 7 >,
   < 1, 6, 10 >,
6  < 2, 4, 3 >,
   < 2, 6, 2 >,
8  < 3, 2, 1 >,
   < 3, 4, 2 >,
10 < 4, 6, 4 >,
   < 5, 3, 3 >,
12 < 5, 4, 3 >,
   < 5, 6, 1 >,
14 < 6, 3, 1 >
   };
16
```

main.mod

```
0  main{
   for (var k=1; k<=2; k++){
2    var source = new IloOplModelSource("lab7.mod");
    var def = new IloOplModelDefinition(source);
4    var opl = new IloOplModel(def, cplex);
    var data = new IloOplDataSource("lab7"+k+".dat");
6    opl.addDataSource(data);
    opl.generate();
8    if (cplex.solve()) {
        var obj=writeln("OBJ = ", cplex.getObjValue());
10    opl.postProcess();
    }
12 }
14 }
```

lab7.mod

```
0  range Nodes = 1..6;
   int file = ...;
2  int B[Nodes]=...;

4  tuple arc {
   key int fromnode;
6   key int tonode;
   float cost;
8  }

10 {arc} Arcs = ...;
```

```

12  dvar int+ x[a in Arcs];
14  minimize
15      sum (a in Arcs)
16          x[a]*a.cost;
18  subject to{
19      forall (i in Nodes)
20          ct:
21              sum (<i,j,c> in Arcs) x[<i,j,c>]
22              - sum (<j,i,c> in Arcs) x[<j,i,c>] == B[i];
23  }
24
25  execute{
26      var ofile=new IloOplOutputFile("lab7"+"_"+file+".txt");
27      ofile.writeln("Objective =", cplex.getObjValue());
28      for (var a in Arcs)
29          ofile.writeln("x", a.fromnode, a.tonode, "=", x[a]);
30      ofile.close();
31  }
32

```

La soluzione della funzione obiettivo riferita al lab7-2.dat : 116

il valore della soluzione ottima

$$\left\{ \begin{array}{l} x_{12} = 12 \\ x_{16} = 0 \\ x_{24} = 0 \\ x_{26} = 0 \\ x_{32} = 0 \\ x_{34} = 8 \\ x_{46} = 0 \\ x_{53} = 0 \\ x_{54} = 2 \\ x_{56} = 10 \\ x_{63} = 0 \end{array} \right.$$

Quindi $\tilde{x}_2 = [12 \ 0 \ 0 \ 0 \ 0 \ 8 \ 0 \ 0 \ 2 \ 10 \ 0]$

La soluzione della funzione obiettivo riferita al file lab7-1.dat: 116

il valore della soluzione ottima

$$\left\{ \begin{array}{l} x_{12} = 12 \\ x_{16} = 0 \\ x_{24} = 2 \\ x_{26} = 0 \\ x_{32} = 0 \\ x_{34} = 8 \\ x_{46} = 0 \\ x_{53} = 0 \\ x_{54} = 0 \\ x_{56} = 10 \\ x_{63} = 0 \end{array} \right.$$

Quindi $\tilde{x}_1 = [12 \ 0 \ 2 \ 0 \ 0 \ 8 \ 0 \ 0 \ 0 \ 10 \ 0]$