Università di Pisa

Fixed Charge Network Flow Problem

Logistics Project Work

Progettare una rete di comunicazione al minimo costo

Prof.ssa Maria Grazia Scutellà

Studente:

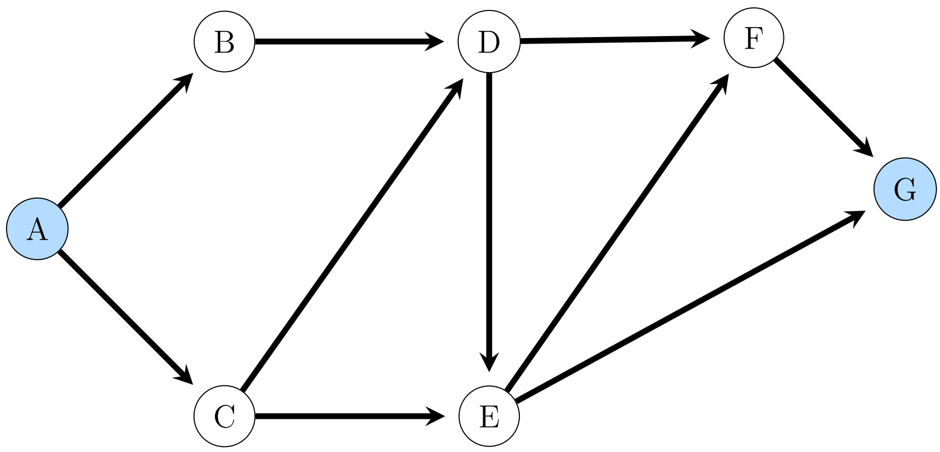
Graziano Amodio

Report

Contents

1. Descrizione del problema

Lo scopo del lavoro è progettare una rete di comunicazione per inviare dati di natura biomedica. I nodi a disposizione {(A, B, C, D, E, F, G)} possono essere interconnessi tramite il seguente insieme di archi orientati:{(A,B), (A,C), (C,E), (B,D), (C,D), (D,E), (D,F), (E,G), (E,F), (F,G)}.



Gli archi hanno un costo fisso di attivazione e un costo variabile a seconda della quantità di banda trasportata. Inoltre gli archi sono capacitati.

La tabella seguente mostra i dati del problema.

Arco	Costo Fisso (Euro)	Costo Variabile (Euro/Mbit/s)	Capacità (Mbit/s)
(A, B)	10,000	300	10
(A, C)	15,000	150	20
(C, E)	30,000	500	10
(B, D)	18,000	300	10
(C, D)	20,000	300	20
(D, E)	12,000	500	7
(D, F)	16,000	350	8
(E, G)	16,000	200	20
(E, F)	12,000	250	17
(F, G)	10,000	300	15

L'obiettivo è valutare quale è il percorso al costo minimo necessario per connettere il cliente in G al database in A, progettando una rete di comunicazione che soddisfi tutti i requisiti presentati precedentemente e che soddisfi la domanda di connettività minimizzando il costo totale sostenuto (costi fissi + costi variabili).

2. Formulazione matematica

Il tipo di problema da risolvere è un *"fixed charge network design problem"*. In particolare in questo caso è necessario progettare un percorso che consideri archi capacitati aventi un costo fisso di attivazione.

2.1 Integer Linear Programming Model

I dati di input sono definiti in questo modo:

- $G = (N, A)$ è un grafo orientato (logistics network);
- b_i è l'equilibrio del nodo $i, \forall i \in N$
- $f_{i,j}$: costi fissi per l'attivazione $\forall (i,j) \in A$
- $c_{i,j}$: costi variabili $\forall (i,j) \in A$
- $u_{i,j}$: capacità degli archi; $\forall (i,j) \in A$
- $x_{i,j}$: quantità di banda trasportata tra gli archi $(i,j), \forall (i,j) \in A$

È necessario introdurre una variabile binaria che ci indichi se un arco è attivato o meno (*design variable*):

$$y_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ è attivato} \\ 0 & \text{altrimenti} \end{cases} \quad , \quad \forall (i,j) \in A$$

Il modello formale di ILP è il seguente:

- Funzione obiettivo:

$$\min \overbrace{\sum_{(i,j) \in A} f_{ij} y_{ij}}^{\text{costi fissi}} + \overbrace{\sum_{(i,j) \in A} c_{ij} x_{ij}}^{\text{costi variabili}}$$

- Soggetta ai seguenti vincoli:

1. **Vincolo di conservazione del flusso**:

$$\sum_{(j,i)\in \mathrm{BS}(i)} x_{ji} - \sum_{(i,j)\in \mathrm{FS}(i)} x_{ij} = b_i \quad \forall i \in N$$

La differenza tra la somma delle quantità entranti e quelle uscenti per ogni nodo è uguale a:

$$b_i = \begin{cases} < 0 & \text{if } i \text{ è nodo di offerta} \\ > 0 & \text{if } i \text{ è nodo di domanda} \\ = 0 & \text{if } i \text{ è nodo di trasporto} \end{cases} , \quad \forall (i,j) \in A$$

2. **Vincolo delle capacità**:

$0 \le x_{ij} \le u_{ij}, \quad \forall (i,j) \in A$

3. **Vincoli di linking**:

$0 \le x_{ij} \le u_{ij}, \quad \forall (i,j) \in A$

4. **Design Variable**:

$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in A$

Andando ad analizzare nel dettaglio i vincoli nel caso specifico:

- Il primo è il vincolo di **conservazione del flusso**.
  Vi è un vincolo per ogni nodo i:

  Da ricordare che: $\sum_{i\in N} b_i = 0$

  (A) $-x_{A,B} - x_{A,C} = -15$
  (B) $x_{A,B} - x_{B,D} = 0$
  (C) $x_{A,C} - x_{C,D} - x_{C,E} = 0$
  (D) $x_{B,D} + x_{C,D} - x_{D,F} - x_{D,E} = 0$
  (E) $x_{C,E} + x_{D,E} - x_{E,G} = 0$
  (F) $x_{D,F} - x_{F,G} = 0$
  (G) $x_{E,G} + x_{F,G} = 15$

- Il secondo vincolo è sulle **capacità**. Vi è un vincolo per ogni arco del grafo:

$$\forall (i,j) \in A : x_{i,j} \leq u_{i,j}$$

$x_{A,B} \leq 10$
$x_{A,C} \leq 20$
$x_{C,E} \leq 10$
$x_{B,D} \leq 10$
$x_{c,D} \leq 20$
$x_{D,E} \leq 7$
$x_{D,F} \leq 8$
$x_{E,G} \leq 20$
$x_{E,F} \leq 17$
$x_{F,G} \leq 15$

- Un terzo vincolo è il **linking constraint**. Vi è un vincolo per ogni arco:

$$\forall (i,j) \in A : x_{i,j} \leq u_{i,j}\, y_{i,j}$$

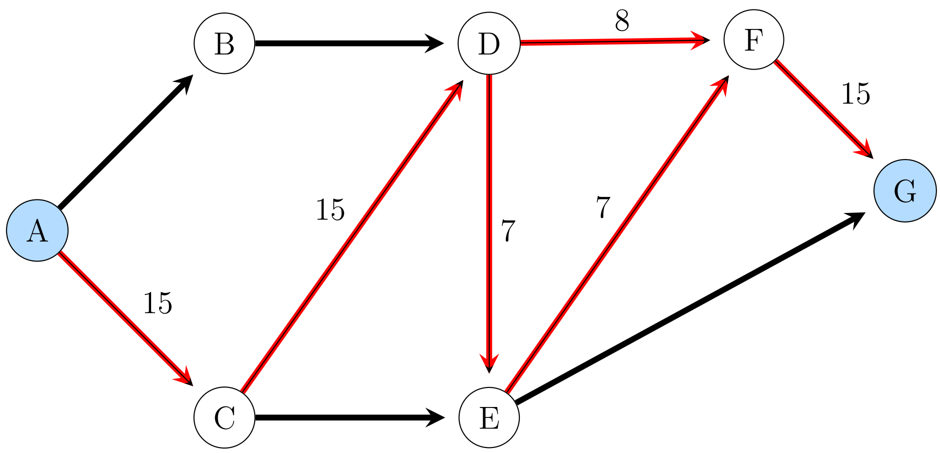
$x_{A,B} \leq 10y_{A,B}$
$x_{A,C} \leq 20y_{A,C}$
$x_{C,E} \leq 10y_{C,E}$
$x_{B,D} \leq 10y_{B,D}$
$x_{c,D} \leq 20y_{c,D}$
$x_{D,E} \leq 7y_{D,E}$
$x_{D,F} \leq 8y_{D,F}$
$x_{E,G} \leq 20y_{E,G}$
$x_{E,F} \leq 7y_{E,F}$
$x_{F,G} \leq 15y_{F,G}$

- Le quantità devono essere: $x_{i,j} \geq 0, \quad \forall (i,j) \in A$
- Una conseguenza che abbiamo è che se l'arco è attivato allora viene trasportata una quantità positiva:

$$y_{ij} = \begin{cases} 1 & \text{if } x_{i,j} > 0 \\ 0 & \text{if } x_{i,j} = 0 \end{cases}, \forall (i,j) \in A$$

2.2 CPLEX result

Il risultato dell'implementazione del problema con AMPL utilizzando come solver il Cplex è mostrato nella figura seguente.



Gli archi attivati sono:
(A,B), (A,C), (B,D), (C,D), (C,E), (D,E), (D,F), (E,F), (E,G), (F,G).
I costi totali sono pari a €104.300,00

2.3 Vincolo del cammino unico

Un altro punto del problema mi chiedeva in imporre un cammino unico al percorso per trasportare la quantità di banda richiesta.

2.4 Integer Linear Programming Model

Conservando tutti i vincoli precedentemente espressi vi è stato aggiunto un altro vincolo:

$$\forall i \in N : \sum_{(i,j)\in A} y_{i,j} \leq 1$$

Per ogni nodo i:
$A : y_{A,B} + y_{A,C} \leq 1$
$C : y_{C,E} + y_{C,D} \leq 1$
$D : y_{D,E} + y_{D,F} \leq 1$
$E : y_{E,F} + y_{E,G} \leq 1$
Nell'esempio del problema in considerazione ho riportato solamente i vincoli dei nodi aventi più di un arco in uscita.

2.5 CPLEX result

Per questo secondo punto non abbiamo una soluzione, infatti considerando le capacità degli archi non è possibile trasportare 15 mb al secondo in quanto alcuni archi sono capacitati ad una quantità minore. Gli archi (D,E), (D,F), (E,F) hanno una capacità massima minore di 15.
Il solver ci restituisce come risultato : *infeasible*

3. Annex

3.1 Mod file

```
set NODES;
set START within (NODES);
set END within (NODES);
set INTERMEDIATE within (NODES);

set EDGES within (NODES cross NODES);

param variable_costs{EDGES};
param capacity{EDGES};
param fixed_costs{EDGES};
param target_capacity;

var Bandwidth{EDGES} >= 0;
var Enabled{EDGES} binary;

# Funzione obiettivo
minimize total_costs:
    sum {(i,j) in EDGES}(fixed_costs[i,j]*Enabled[i,j]+variable_costs[i,j]*Bandwidth[i,j]);

# Capacity constraint
subject to Capacity_Constraint {(i,j) in EDGES}:
    Bandwidth[i,j] <= capacity[i,j];

# Conservazione del flow
subject to Start_Flow {s in START}:
    -sum {(s,n) in EDGES} Bandwidth[s,n] == -target_capacity;
subject to Intermediate_Flow {i in INTERMEDIATE}:
    sum {(n,i) in EDGES} Bandwidth[n,i] - sum {(i,m) in EDGES} Bandwidth[i,m] == 0;
subject to End_Flow {e in END}:
    sum {(n,e) in EDGES} Bandwidth[n,e] == target_capacity;

# Linking constraint
subject to Link_Capacity {(i,j) in EDGES}:
    Bandwidth[i,j] <= capacity[i,j]*Enabled[i,j];
```

Figure 1: biomed.mod

Di seguito il file .mod della variante con il cammino unico:

biomed.mod | biomed_var.mod

```
set NODES;
set START within (NODES);
set END within (NODES);
set INTERMEDIATE within (NODES);

set EDGES within (NODES cross NODES);

param variable_costs{EDGES};
param capacity{EDGES};
param fixed_costs{EDGES};
param target_capacity;

var Bandwidth{EDGES} >= 0;
var Enabled{EDGES} binary;

# Funzione obiettivo
minimize total_costs:
    sum {(i,j) in EDGES}(fixed_costs[i,j]*Enabled[i,j]+variable_costs[i,j]*Bandwidth[i,j]);

# Conservazione del flow
subject to Start_Flow {s in START}:
    -sum {(s,n) in EDGES} Bandwidth[s,n] == -target_capacity;
subject to Intermediate_Flow {i in INTERMEDIATE}:
    sum {(n,i) in EDGES} Bandwidth[n,i] - sum {(i,m) in EDGES} Bandwidth[i,m] == 0;
subject to End_Flow {e in END}:
    sum {(n,e) in EDGES} Bandwidth[n,e] == target_capacity;

# Capacity constraint
subject to Link_Capacity {(i,j) in EDGES}:
    Bandwidth[i,j] <= capacity[i,j]*Enabled[i,j];

# Unique path constraint
subject to Unique_Path {a in NODES}:
    sum {(a,k) in EDGES} Enabled[a,k] <= 1;
```

Figure 2: biomed variante.mod

3.2 Dat file

```
data;

set NODES := A, B, C, D, E, F, G;
set START := A;
set END := G;
set INTERMEDIATE := B, C, D, E, F;

set EDGES :=
(A,B)
(A,C)
(C,E)
(B,D)
(C,D)
(D,E)
(D,F)
(E,G)
(E,F)
(F,G);

param: variable_costs, capacity, fixed_costs :=
A,B    300             10         10000
A,C    150             20         15000
C,E    500             10         30000
B,D    300             10         18000
C,D    300             20         20000
D,E    500             7          12000
D,F    350             8          16000
E,G    200             20         16000
E,F    250             17         12000
F,G    300             15         10000;

param target_capacity := 15;
```

Figure 3: biomed.dat

3.3 Run file

```
biomed.dat    biomed.run

reset;
model biomed_var.mod;
data biomed.dat;
option solver cplex;

for {i in 1..2}{
  if i == 1 then {
    drop Unique_Path;
    printf "SOLUZIONE AL PUNTO 2\n";
  }
  if i == 2 then {
    restore Unique_Path;
    printf "SOLUZIONE AL PUNTO 4\n";
  }

  solve;

  if solve_result == "solved" then{
    display Bandwidth;
    printf "Costi totali: %f Euro\n", total_costs;
  }
  else {
    printf "La soluzione non é stata trovata. Codice solve_result: %s\n", solve_result;
  }
  printf "\n------------------------------------------\n\n";
}
```

Figure 4: biomed.run

3.4 Soluzione

```
AMPL
ampl: include biomed.run;
SOLUZIONE AL PUNTO 2
CPLEX 22.1.1.0: optimal integer solution; objective 104300
10 MIP simplex iterations
0 branch-and-bound nodes
Bandwidth :=
A B     0
A C    15
B D     0
C D    15
C E     0
D E     7
D F     8
E F     7
E G     0
F G    15
;

Costi totali: 104300.000000 Euro

------------------------------------------

SOLUZIONE AL PUNTO 4
presolve, constraint Link_Capacity['D','F']:
        all variables eliminated, but upper bound = -8 < 0
La soluzione non é stata trovata. Codice solve_result: infeasible

------------------------------------------
```

Figure 5: soluzione