Azzi Elena (matricola 358514)
Cella Arianna (matricola 364637)

Cammino minimo per agenti multipli

Testo del problema

Sia dato un grafo non orientato con archi aventi tutti tempo di percorrenza pari a 1. Su questo grafo si muovono n agenti. Ogni agente *i* ha nodo di origine o_i e un nodo destinazione d_i. Si definisca un modello matematico per minimizzare la somma dei cammini minimi di tutti gli agenti, tenendo conto che gli agenti non possono trovarsi nello stesso istante in un nodo o lungo un arco.

Si formuli il modello matematico in AMPL e si definiscano i dati di una particolare istanza, risolvendola. Si faccia inoltre un'analisi di cosa succede se si modificano alcuni dei dati dell'istanza.

Modello matematico e traduzione in AMPL

In primo luogo, è stato necessario definire gli **insiemi** del nostro problema, ovvero l'insieme dei nodi del grafo (INTER), l'insieme degli agenti e l'insieme degli archi del grafo (ROADS) che collegano due elementi appartenenti all'insieme dei nodi.

$$set\ INTER;$$
$$set\ AGENTI;$$
$$set\ ROADS\ within\ (INTER\ cross\ INTER);$$

I **parametri** che abbiamo scelto per la risoluzione sono il nodo origine o_i dal quale l'agente parte, il nodo destinazione d_i che rappresenta il nodo che l'agente deve raggiungere ed infine $Tmax$ che rappresenta il tempo massimo entro il quale tutti gli agenti devono raggiungere la loro rispettiva destinazione.

$$param\ entr\{AGENTI\}\ symbolic\ in\ INTER;$$
$$param\ exit\{AGENTI\}\ symbolic\ in\ INTER;$$
$$param\ Tmax;$$

La **variabile decisionale** che abbiamo usato per risolvere il problema è $Use_{i,j,k,t}$, una variabile binaria che indica se l'agente k attraversa l'arco (i,j) all'istante di tempo t.

$$Use_{i,j,k,t} = \begin{cases} 1\ se\ l'arco\ (i,j)\ \text{è}\ percorso\ dall'agente\ k\ al\ tempo\ t \\ 0\ se\ l'arco\ (i,j)\ non\ \text{è}\ percorso\ dall'agente\ k\ al\ tempo\ t \end{cases}$$

$$var\ Use\ \{(i,j) in\ ROADS, k\ in\ AGENTI, t\ in\ 1..Tmax\}\ binary;$$

L'**obiettivo** del problema è minimizzare la somma dei cammini minimi di tutti gli agenti

$$\text{Min} \sum_{t\in\{1,\dots,Tmax\}} \sum_{k\in AGENTI} \sum_{(i,j)\in ARCHI} Use_{i,j,k,t} - \sum_{t\in\{1,\dots,Tmax\}} \sum_{k\in AGENTI} Use_{exit[k],exit[k],k,t}$$

$$minimize\ Total\_Time: (sum\ \{(i,j)\ in\ ROADS, k\ in\ AGENTI, t\ in\ 1..Tmax\}\ \ Use[i,j,k,t]$$
$$-\ sum\{\ k\ in\ AGENTI, t\ in\ 1..Tmax\}\ Use[exit[k], exit[k], k, t];$$

I **vincoli** che definiscono la regione ammissibile del problema sono:

1. Il percorso dell'agente deve iniziare dal nodo di origine, ovvero la sommatoria degli archi uscenti dal nodo origine di un agente k al tempo $t = 1$ deve essere pari a 1

$$\sum_{(entr[k],j)\,\in\,ROADS} Use_{entr[k],j,k,1} = 1 \quad \forall k \in AGENTI$$

$$subject\ to\ Start\ \{k\ in\ AGENTI\}\colon sum\ \{(entr[k],j)\ in\ ROADS\}\ Use[entr[k],j,k,1]\ =\ 1;$$

2. Il percorso dell'agente deve terminare nel nodo di destinazione, ovvero la sommatoria degli archi entranti nel nodo destinazione di un agente k al tempo $t = Tmax$ deve essere pari a 1

$$\sum_{(i,exit[k])\,\in\,ROADS} Use_{i,exit[k],k,Tmax} = 1 \quad \forall k \in AGENTI$$

$$subject\ to\ End\ \{k\ in\ AGENTI\}\colon sum\ \{(j,exit[k])\ in\ ROADS\}\ Use[j,exit[k]\,,k,Tmax]\ =\ 1;$$

3. Due o più agenti non possono sovrapporsi al tempo t nello stesso nodo i, ovvero la sommatoria degli archi entranti nel nodo i al tempo t, considerando tutti gli agenti, deve essere al massimo pari 1. Questo significa che al massimo un agente può entrare allo stesso istante di tempo nello stesso nodo, garantendo così la non sovrapposizione.

$$\sum_{k\,\in\,AGENTI}\ \sum_{(i,j)\,\in\,ROADS} Use_{i,j,k,t} \le 1 \qquad \forall i \in INTER, \forall t \in \{1,\dots,Tmax\}$$

$$subject\ to\ NonSovrapposizioneNodi\ \{i\ in\ INTER, t\ in\ 1..Tmax\}\colon$$
$$sum\ \{k\ in\ AGENTI,(i,j)\ in\ ROADS\}\ Use[i,j,k,t]\ \ <=\ 1;$$

4. Due o più agenti non possono sovrapporsi al tempo t nello stesso arco non orientato (i,j), ovvero essendo che due agenti non possono trovarsi nello stesso nodo contemporaneamente non si vuole che ci siano allo stesso istante di tempo l'agente k nell'arco (i,j) e l'agente h nell'arco (j,i) essendo un grafo non orientato.

$$Use_{i,j,k,t} + Use_{j,i,h,t} \le 1 \qquad \forall (i,j) \in ROADS, \forall t \in \{1,\dots,Tmax\}, \forall k,h \in AGENTI\ :\ h \ne k$$

$$subject\ to\ NonSovrapposizioneArchi\ \{k\ in\ AGENTI, h\ in\ AGENTI, t\ in\ 1..Tmax, (i,j)\ in\ ROADS:$$
$$h\,!=\ k\ \&\&\ i <= j\}: Use[i,j,k,t]\ +\ Use[j,i,h,t]\ \ <=\ 1;$$

5. L'agente può viaggiare solamente verso nodi direttamente collegati al nodo da cui è partito nell'istante temporale precedente.

$$\sum_{(i,j)\in ROADS} Use_{i,j,k,t} = \sum_{(m,i)\in ROADS} Use_{m,i,k,(t-1)} \quad \forall k \in AGENTI, \forall t \in 2,\dots,Tmax, \forall i \in INTER$$

$$subject\ to\ NodiAdiacenti\ \{k\ in\ AGENTI, t\ in\ 2..Tmax, i\ in\ INTER\}\colon$$
$$sum\ \{(i,j)\ in\ ROADS\}\ Use[i,j,k,t]\ =\ sum\ \{(m,i)\ in\ ROADS\}\ Use[m,i,k,t-1];$$

In sintesi, il **modello matematico** è il seguente:

$$\text{Min} \sum_{t \in \{1,\dots,Tmax\}} \sum_{k \in AGENTI} \sum_{(i,j) \in ARCHI} Use_{ijkt} - \sum_{t \in \{1,\dots,Tmax\}} \sum_{k \in AGENTI} Use_{exit[k],exit[k],k,t}$$

$$\sum_{(entr[k],j) \in ROADS} Use_{entr[k],j,k,1} = 1 \;\; \forall k \in AGENTI$$

$$\sum_{(i,exit[k]) \in ROADS} Use_{i,exit[k],k,Tmax} = 1 \quad \forall k \in AGENTI$$

$$\sum_{k \in AGENTI} \sum_{(i,j) \in ROADS} Use_{i,j,k,t} \leq 1 \qquad \forall i \in INTER, \forall t \in \{1,\dots,Tmax\}$$

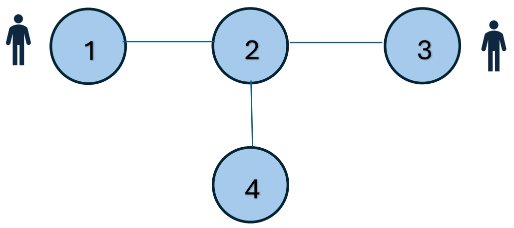
$$Use_{i,j,k,t} + Use_{j,i,h,t} \leq 1 \qquad \forall (i,j) \in ROADS, \forall t \in \{1,\dots,Tmax\}, \forall k,h \in AGENTI : \; h \neq k$$

$$\sum_{(i,j) \in ROADS} Use_{i,j,k,t} = \sum_{(m,i) \in ROADS} Use_{m,i,k,(t-1)} \;\; \forall k \in AGENTI, \forall t \in 2,\dots,Tmax, \forall i \in INTER$$

$$Use_{i,j,k,t} \in \{0,1\}$$

Risoluzione di problemi

Problema 1: L'agente 1 si trova nel nodo 1 e vuole arrivare al nodo 3. L'agente 2 si trova nel nodo 3 deve vuole arrivare nel nodo 1.



Problema1.dat

```
set INTER := 1 2 3 4;
set ROADS :=
(1, 1)
(1, 2)
(2, 1)
(2, 2)
(2, 3)
(3, 2)
(3, 3)
(4, 4)
(2, 4)
(4, 2);
set AGENTI := a1 a2;

param: entr   exit :=
    a1   1    3
    a2   3    1;
param Tmax :=10;
```

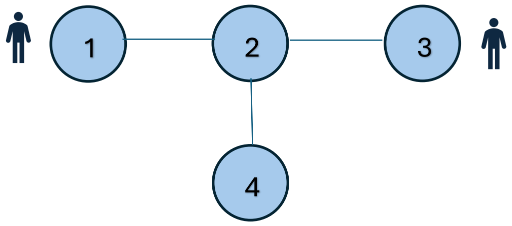
Risultati

```
Gurobi 4.0.1: optimal solution;
objective 7
50 simplex iterations
Percorso degli agenti:
Agente a1:
Tempo 1:   Nodo 1 -> Nodo 2
Tempo 2:   Nodo 2 -> Nodo 4
Tempo 3:   Nodo 4 -> Nodo 2
Tempo 4:   Nodo 2 -> Nodo 3

Agente a2:
Tempo 1:   Nodo 3 -> Nodo 3
Tempo 2:   Nodo 3 -> Nodo 2
Tempo 3:   Nodo 2 -> Nodo 1

[viene visualizzato fino al
tempo Tmax che entrambi gli
agenti rimangono sul loro nodo
destinazione]
```

Azzi Elena (matricola 358514)
Cella Arianna (matricola 364637)

Problema 2:

L'agente 1 si trova nel nodo 1 e vuole arrivare al nodo 2. L'agente 2 si trova nel nodo 3 e vuole arrivare nel nodo 1.



Problema2.dat

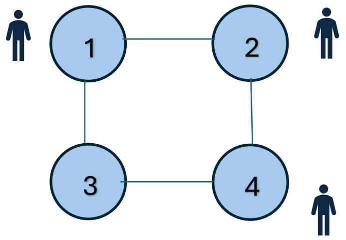
```
set INTER := 1 2 3 4;
set ROADS :=
(1, 1) (1, 2)
(2, 1) (2, 2)
(2, 3) (3, 2)
(3, 3) (4, 4)
(2, 4) (4, 2);
set AGENTI := a1 a2;

param: entr   exit :=
    a1    1    2
    a2    3    1;
param Tmax :=10;
```

Risultati

```
Gurobi 4.0.1: optimal solution;
objective 6
50 simplex iterations
Percorso degli agenti:
Agente a1:
Tempo 1:   Nodo 1 -> Nodo 2
Tempo 2:   Nodo 2 -> Nodo 4
Tempo 3:   Nodo 4 -> Nodo 2

Agente a2:
Tempo 1:   Nodo 3 -> Nodo 3
Tempo 2:   Nodo 3 -> Nodo 2
Tempo 3:   Nodo 2 -> Nodo 1
```

Problema 3:

L'agente 1 si trova nel nodo 4 e vuole arrivare al nodo 1. L'agente 2 si trova nel nodo 2 e vuole arrivare nel nodo 3. L'agente 3 si trova nel nodo 1 e vuole arrivare nel nodo 4.



Problema3.dat

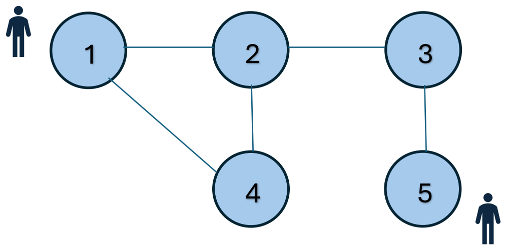
```
set INTER := 1 2 3 4;
set ROADS :=
(1, 1)
(2, 2)
(3, 3)
(4, 4)
(1, 2)
(2, 1)
(2, 4)
(4, 2)
(3, 4)
(4, 3)
(1, 3)
(3, 1);
set AGENTI := a1 a2 a3;

param: entr   exit :=
    a1    4    1
    a2    2    3
    a3    1    4;

param Tmax :=4;
```

Risultati

```
Gurobi 4.0.1: optimal solution;
objective 6
24 simplex iterations
Percorso degli agenti:
Agente a1:
Tempo 1:   Nodo 4 -> Nodo 2
Tempo 2:   Nodo 2 -> Nodo 1
Tempo 3:   Nodo 1 -> Nodo 1
Tempo 4:   Nodo 1 -> Nodo 1

Agente a2:
Tempo 1:   Nodo 2 -> Nodo 1
Tempo 2:   Nodo 1 -> Nodo 3
Tempo 3:   Nodo 3 -> Nodo 3
Tempo 4:   Nodo 3 -> Nodo 3

Agente a3:
Tempo 1:   Nodo 1 -> Nodo 3
Tempo 2:   Nodo 3 -> Nodo 4
Tempo 3:   Nodo 4 -> Nodo 4
Tempo 4:   Nodo 4 -> Nodo 4
```

Problema 4:

L'agente 1 si trova nel nodo 1 e vuole arrivare al nodo 3. L'agente 2 si trova nel nodo 5 e vuole arrivare nel nodo 1.



Problema4.dat

```
set INTER := 1 2 3 4 5;
set ROADS :=
(1, 1) (1, 2) (2, 1)
(2, 2) (2, 3) (3, 2)
(3, 3) (4, 4) (2, 4)
(4, 2) (1, 4) (4, 1)
(3, 5) (5, 3) (5, 5);
set AGENTI := a1 a2;

param: entr   exit :=
    a1   1   3
    a2   5   1;

param Tmax := 4;
```

Risultati

```
Gurobi 4.0.1: optimal solution;
objective 7
58 simplex iterations
Percorso degli agenti:
Agente a1:
Tempo 1:   Nodo 1 -> Nodo 2
Tempo 2:   Nodo 2 -> Nodo 4
Tempo 3:   Nodo 4 -> Nodo 2
Tempo 4:   Nodo 2 -> Nodo 3

Agente a2:
Tempo 1:   Nodo 5 -> Nodo 3
Tempo 2:   Nodo 3 -> Nodo 2
Tempo 3:   Nodo 2 -> Nodo 1
Tempo 4:   Nodo 1 -> Nodo 1
```

Analisi dei risultati

Il modello è in grado di risolvere i problemi descritti efficacemente, producendo il miglior risultato possibile. Tuttavia, se si modifica il valore di $Tmax$ impostandolo a un valore inferiore al minimo necessario, il problema diventa irrisolvibile e il modello restituirà "$infeasible$" come risultato.

Se si modificano gli archi non orientati del nostro grafo, si alterano di conseguenza i possibili percorsi che un agente potrebbe seguire per arrivare al proprio obiettivo, influenzando quindi la soluzione complessiva del modello.

L'aggiunta o la rimozione di nodi modifica la struttura del grafo e, di conseguenza, l'insieme degli archi che li collegano. Se vengono aggiunti nodi, può esserci una maggiore flessibilità nel trovare la soluzione migliore; al contrario, se si avesse la rimozione di un nodo si potrebbe avere una limitazione dei percorsi disponibili portando anche ad una situazione "$infeasible$".

Aumentare o diminuire il numero di agenti può influenzare la congestione della rete del grafo e quindi le soluzioni ottimali, portando a tempi di percorrenza più lunghi o ad una situazione "$infeasible$".

Infine, anche se questa situazione non deve accadere se si segue il testo del problema, rilassando i vincoli che limitano il numero degli agenti presenti in un determinato istante di tempo su un nodo o lungo un arco (ad esempio, permettendo che al massimo due agenti si possano trovare in un determinato nodo o su un determinato arco) permetterebbe di ottenere risultati uguali o migliori rispetto alla restrizione richiesta dal testo del problema.

È importante notare che a causa delle limitazioni computazionali della versione gratuita di AMPL, non sempre è possibile testare il modello su problemi di modeste dimensioni. Pertanto, la scalabilità del modello potrebbe essere limitata quando si affrontano scenari più complessi.