

# PROGETTO AMPL

Matteo Di Giorgio, Matricola: 353719

September 1, 2023

## 1 CAMMINO MINIMO PER AGENTI MULTIPLI

Il modello matematico descritto è un problema di routing multi-agente, in cui un insieme di agenti deve essere indirizzato da un nodo di partenza a un nodo di destinazione attraverso un insieme di archi evitando le collisioni tra essi.

In questo caso studio si ipotizza che ogni agente non può sostare su un nodo.

### 1.1 Modello

Abbreviazioni: LP = LimitePercorso, ARCHIB = ARCHI BIDIREZIONALI, ArcU = ArcoUtilizzato

$$\text{minimize} \quad \sum_{a \in \text{Agenti}} \sum_{t \in \text{LP}} \sum_{(i,j) \in \text{ARCHIB}} \text{Arc}U_{a,t,(i,j)}$$

s.t.

$$\text{Agenti} > 0$$

$$\text{Origine}_a \in \text{NODI}, \quad \forall a \in \text{Agenti}$$

$$\text{Destinazione}_a \in \text{NODI}, \quad \forall a \in \text{Agenti}$$

$$\text{LP} > 0$$

$$\text{Arc}U_{a,t,(i,j)} \in 0, 1 \quad \forall a \in \text{Agenti}, \forall t \in \text{LP}, \forall (i,j) \in \text{ARCHIB}$$

$$\sum_{(i,j) \in \text{ARCHIB}: i = \text{Origine}_a} \text{Arc}U_{a,1,(i,j)} = 1 \quad \forall a \in \text{Agenti}$$

$$\sum_{(i,k) \in \text{ARCHIB}} \text{Arc}U_{a,t,(i,k)} = \sum_{(k,j) \in \text{ARCHIB}} \text{Arc}U_{a,t+1,(k,j)} \quad \forall a \in \text{Agenti}, \forall t \in \text{LP}, \forall k \in \text{NODI}$$

$$\text{with } (t \neq 1 | k \neq \text{Origine}_a) \quad \text{and} \quad (t = 1 | k \neq \text{Destinazione}_a) \quad \text{and} \quad t \neq \text{LP}$$

$$\sum_{a \in \text{Agenti}} \sum_{(i,k) \in \text{ARCHIB}} \text{Arc}U_{a,t,(i,k)} \leq 1 \quad \forall (k,t) | \forall k \in \text{NODI}, \forall t \in \text{LP}$$

$$\text{Arc}U_{a,t,(i,j)} + \sum_{a2 \in \text{Agenti}: a2 \neq a} \text{Arc}U_{a2,t,(j,i)} \leq 1 \quad \forall a \in \text{Agenti}, \forall t \in \text{LP}, \forall (i,j) \in \text{ARCHIB}$$

## 1.2 Insiemi

- **NODI:** Indica l'insieme dei nodi.
- **ARCHI:** Indica l'insieme degli archi.
- **ARCHI BIDIREZIONALI:** Indica che il set di ARCHI è utilizzabile in entrambe le direzioni costituendo un nuovo set.

## 1.3 Parametri

- **Origine:** Indica il nodo di origine di un determinato agente.
- **Destinazione:** Indica il nodo di destinazione di un determinato agente.
- **Agenti:** Indica un numero maggiore di 0 di agenti da inserire nel grafo.
- **Limite Percorso:** Indica un limite massimo di istanti temporali per un percorso.

## 1.4 Variabili

- **ArcoUtilizzato:** Variabile binaria che indica se l'arco è utilizzato da un agente nel suo percorso minimo.

## 1.5 Obiettivo

L'obiettivo del modello è minimizzare il costo totale del percorso di tutti gli agenti, ovvero la somma dei costi degli archi attraversati da ciascun agente.

## 1.6 AMPL

### 1.6.1 Vincoli

- **Il primo vincolo *nodo Partenza*:** assicura che all'istante temporale  $t=1$  ogni agente parte dal suo nodo di partenza.
- **il secondo vincolo *vincolo Flusso*:** assicura che quando in un istante temporale  $t$ ,  $n$  agenti entrano in un nodo, allora all'istante  $t+1$   $n$  agenti usciranno da quel nodo. Inoltre sono gestiti internamente gli istanti temporali e la relativa posizione dell'agente in quel determinato istante.
- **Il terzo vincolo *overlap Agenti*:** definisce che all'istante temporale definito  $n$  agenti non posso percorrere lo stesso arco che finirebbe sullo stesso nodo.
- **Il quarto vincolo *collisioni Agenti*:** assicura se un agente utilizza un arco  $(i,j)$  nell'istante  $t$ , un altro agente non potrà usare l'arco  $(j,i)$  al medesimo istante.

### 1.6.2 Traduzione vincoli in AMPL

```
subject to nodo_Partenza{a in 1..Agenti}:  
    sum{(i,j) in ARCHI_BIDIREZIONALI: i = Origine[a]} ArcoUtilizzato[a,1, i, j] = 1;
```

Figure 1: Traduzione primo vincolo

```
subject to vincolo_Flusso{a in 1..Agenti, t in 1..Limite_Percorso, k in NODI:(t!=1 || k!=Origine[a]) && (t==1 || k!=Destinazione[a]) && t!=Limite_Percorso}:  
    sum{(i, k) in ARCHI_BIDIREZIONALI} ArcoUtilizzato[a,t,i,k] = sum{(k, j) in ARCHI_BIDIREZIONALI} ArcoUtilizzato[a,t+1, k, j];
```

Figure 2: Traduzione secondo vincolo

```

subject to overlap_Agenti{(k,t) in NODI cross (1..Limite_Percorso)}:
    sum{a in 1..Agenti} sum{(i,k) in ARCHI_BIDIREZIONALI} ArcoUtilizzato[a,t,i,k] <= 1;

```

Figure 3: Traduzione terzo vincolo

```

subject to collisioni_Agenti{a in 1..Agenti, t in 1..Limite_Percorso, (i,j) in ARCHI_BIDIREZIONALI}:
    ArcoUtilizzato[a,t,i,j] + sum{a2 in 1..Agenti: a2!=a} ArcoUtilizzato[a2,t,j,i] <= 1;

```

Figure 4: Traduzione quarto vincolo

### 1.6.3 Esempio

Agente1 va dal nodo A al nodo D, Agente2 va dal nodo D al nodo B, Agente3 va dal nodo C al nodo B

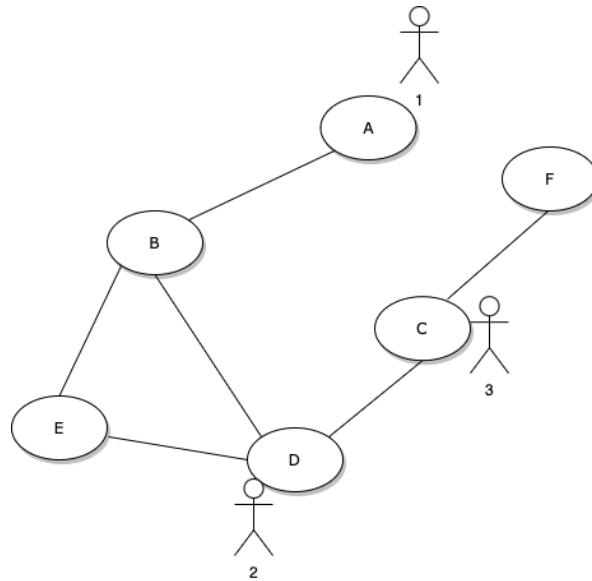


Figure 5: Grafo istanze esempio

### 1.6.4 Risultato

```

Istante temporale 1
  agente: 1 -> da A a B.
Istante temporale 1
  agente: 2 -> da D a E.
Istante temporale 1
  agente: 3 -> da C a D.

Istante temporale 2
  agente: 1 -> da B a D.
Istante temporale 2
  agente: 2 -> da E a B.
Istante temporale 2
  agente: 3 -> da D a E.

Istante temporale 3
  agente: 3 -> da E a B.

```

Figure 6: Percorsi agenti calcolati dall'algoritmo

Il risultato dell'obiettivo risulta essere 7 come la somma degli archi attraversati dagli agenti nei loro percorsi minimi.

## 1.7 Modifiche alle istanze

Se si modificano alcuni dati dell'istanza del problema possono verificarsi diverse conseguenze. Ecco alcune possibili modifiche e il loro impatto sul problema:

- **Modifica del numero di agenti:** Se il numero di agenti aumenta, il costo totale del percorso potrebbe aumentare in modo significativo. Al contrario, se il numero di agenti diminuisce, il costo totale del percorso potrebbe diminuire, ma potrebbe anche verificarsi il caso in cui alcuni agenti debbano coprire maggiori distanze e quindi il costo totale del percorso potrebbe aumentare.
- **Modifica delle posizioni delle destinazioni:** Se le posizioni delle destinazioni cambiano, il percorso minimo potrebbe cambiare. Potrebbe verificarsi il caso in cui alcune destinazioni siano ora più vicine tra loro e quindi il costo totale del percorso diminuisca. Al contrario, potrebbe anche verificarsi il caso in cui le destinazioni siano ora più distanti tra loro e quindi il costo totale del percorso aumenti.

In sintesi, le modifiche dei dati dell'istanza del problema possono avere un impatto significativo sulla soluzione ottimale. Pertanto, è importante considerare attentamente le modifiche ai dati dell'istanza e valutare le loro conseguenze prima di risolvere il problema.

Un'ulteriore osservazione si può fare se si modificano i dati impostati dal problema stesso come:

- **Modifica del costo delle distanze:** Se il costo delle distanze tra le destinazioni cambia, il percorso minimo potrebbe cambiare. Potrebbe verificarsi il caso in cui alcune distanze diventino meno costose e quindi il costo totale del percorso diminuisca. Al contrario, potrebbe anche verificarsi il caso in cui alcune distanze diventino più costose e quindi il costo totale del percorso aumenti.
- **Modifica della capacità degli agenti:** Se la capacità degli agenti cambia, potrebbe essere necessario modificare il numero di agenti o il percorso per garantire che tutte le destinazioni vengano coperte. Potrebbe verificarsi il caso in cui la capacità degli agenti aumenti e quindi sia possibile coprire più destinazioni con lo stesso numero di agenti. Al contrario, potrebbe anche verificarsi il caso in cui la capacità degli agenti diminuisca e quindi sia necessario utilizzare più agenti per coprire tutte le destinazioni.

### 1.7.1 Caso Studio

In questo esempio viene messa in risalto la capacità del modello al cercare percorsi alternativi in caso di conflitti. Modificando l'arco (C,D) con l'arco (B,F) si ottengono percorsi alternativi (si ricordi che un agente non può sostare sul nodo):

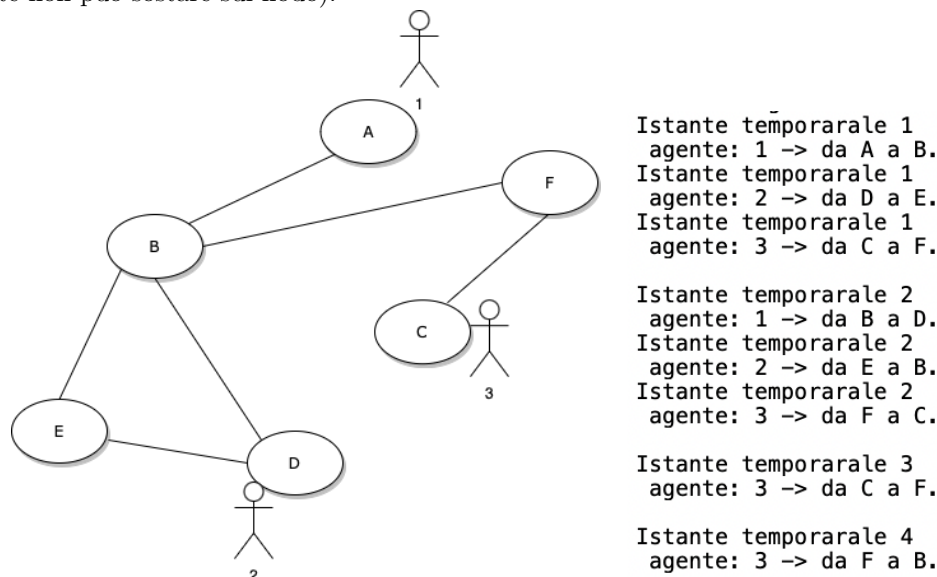


Figure 7: Grafo modificato e Percorsi