1 Lezione del 22-10-24

1.1 Problema del commesso viaggiatore

Problema 1.1: del commesso viaggiatore

Supponiamo che un commesso viaggiatore debba fare il giro di 5 città, passando da tutte una e una sola volta. Si prepara una tabella con le distanze fra le città:

	Roccalbegna	Cana	Vallerona	Santa Caterina	Triana
Roccalbegna	-	18	14	17	19
Cana	16	-	19	22	23
Vallerona	17	14	-	18	20
Santa Caterina	16	19	22	-	21
Triana	15	14	13	20	-

Quale percorso dovrà seguire il commerciante, in modo da minimizzare la distanza percorsa?

Il problema del commesso viaggiatore (in inglese *Traveling Salesman Problem*, TSP) è effettivamente quello di trovare un **ciclo hamiltioniano**, cioè ciclo su un grafo che passa da ogni nodo una e una sola volta. Possiamo innanzitutto porre la matrice di adiacenza:

$$C = \begin{pmatrix} - & 18 & 14 & 17 & 19 \\ 16 & - & 19 & 22 & 23 \\ 17 & 14 & - & 18 & 20 \\ 16 & 19 & 22 & - & 21 \\ 15 & 14 & 13 & 20 & - \end{pmatrix}$$

Notiamo che si può distinguere fra TSP **simmetrici** e **asimmetrici**, in base alla simmetria della matrice di adiacenza. Si ha che i due tipi di problema hanno algoritmi risolutivi molto diversi. In questo caso, come è chiaro dalla matrice, considereremo il caso **asimmetrico**.

Inoltre, senza togliere dalla generalità della trattazione, possiamo assumere tutte le connessioni come stabilite, quindi la matrice **completa**. In questo caso gli archi mancanti del grafo saranno rappresentati da un costo infinito nella matrice di adiacenza (compresi gli archi $x_{ij} \to x_{ij}$, cioè da un nodo allo stesso nodo).

Abbiamo che tutti i percorsi possibili, per n città partendo dalla prima, sono $(n-1)!$, e che il problema si dimostra NP-completo. Diventa inapplicabile un'algoritmo di enumerazione completa: dobbiamo trovare quindi un modello matematico su cui applicare i macchinari della PL. Chiamiamo C la matrice dei costi, a_{ij} l'arco generico, C_H il ciclo hamiltioniano e introduciamo la variabile binaria:

$$x_{ij} = \begin{cases} 0, & a_{ij} \notin C_H \\ 1, & a_{ij} \in C_H \end{cases}$$

che rappresenterà matrice di adiacenza rappresentante il percorso scelto. Come sempre, linearizzamo le matrici in vettori ordinati lessicograficamente, quindi:

$$C = \begin{pmatrix} x_{11} & x_{12} & ... & x_{1N} \\ x_{21} & x_{22} & ... & x_{2N} \\ ... \\ x_{N1} & x_{N2} & ... & x_{NN} \end{pmatrix} \to c = (x_{11}, x_{12}, ..., x_{1N}, x_{21}, x_{22}, ..., x_{2N}, ..., x_{N1}, x_{N2}, ..., x_{NN})$$

A questo punto possiamo impostare il problema di ILP:

$$\begin{cases} \min c^T x \\ x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 1 \\ x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 1 \\ ... \\ x_{51} + x_{52} + x_{53} + x_{54} + x_{55} = 1 \\ x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1 \\ x_{12} + x_{22} + x_{32} + x_{42} + x_{52} = 1 \\ .... \\ x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1 \\ x \in \{0, 1\} \end{cases}$$

I primi $n = 5$ vincoli impongono che da ogni nodo esca uno e uno solo arco, mentre i seguenti $n = 5$ impongono che su ogni nodo arrivi uno e un solo arco. L'ultimo vincolo rappresenta il dominio booleano delle variabili (un arco o è incluso o non è incluso nel ciclo hamiltoniano).

Notiamo che questo non basta a rappresentare cicli hamiltoniani: sono infatti ammessi sottocicli disgiunti fra di loro. Aggiungiamo quindi i vincoli, detti **di connessione**, dato $S \subset N$ sottoinsieme qualsiasi dei nodi:

$$\sum_{i \in S \, j \notin S} x_{ij} \geq 1, \quad \forall S \subset N, \quad S \neq \emptyset$$

Questi vincoli rappresentano l'obbligo per ogni sottoinsieme S non vuoto di nodi di avere almeno un'arco uscente, così da evitare cicli isolati. Si prende sottoinsieme stretto in quanto sarebbe inutile chiedere un'arco uscente dall'insieme completo dei nodi. Notiamo inoltre che con $|S| = 1$, questo dà i primi vincoli del problema già posto (in forma $x_{11} + x_{12} + x_{13}...$), che risulta in quanto sarebbero gli archi uscenti da gruppi di cardinalità 1, cioè singoli nodi, e il singolo arco uscente da ogni singolo nodo è già una prerogativa dell'assegnamento.

Possiamo quindi dire, in modo più completo, che si prendono i vincoli:

$$\begin{cases} \sum_{j \in N, j \neq i} x_{ij} = 1, \quad \forall i \in N \\ \sum_{i \in N, i \neq j} x_{ij} = 1, \quad \forall j \in N \\ \sum_{i \in S, j \notin S} x_{ij} \geq 1, \forall S \subset N, \quad 2 \leq |S| \leq n - 1 \end{cases}$$

1.1.1 Cardinalità dei vincoli

Si ha che i vincoli iniziali erano $2n$, n per gli archi entranti e n per gli archi uscenti. Inoltre, posto $2 \leq |S| \leq n - 1$, abbiamo che il numero di vincoli di connessione è:

$$|\mathcal{V}_{connessione}| = 2^n - n - 2$$

Ergo si ha un numero di vincoli pari a:

$$|\mathcal{V}| = 2^n - n - 2 + 2n = 2^n + n - 2$$

Si ha che i vincoli crescono quindi in maniera esponenziale. Ad esempio, ecco l'insieme completo dei vincoli preso $n = 5$, come dall'esempio precedente, generati attraverso un programma al computer:

$$\begin{cases}
x_{1\,2} + x_{1\,3} + x_{1\,4} + x_{1\,5} = 1 \\
x_{2\,1} + x_{2\,3} + x_{2\,4} + x_{2\,5} = 1 \\
x_{3\,1} + x_{3\,2} + x_{3\,4} + x_{3\,5} = 1 \\
x_{4\,1} + x_{4\,2} + x_{4\,3} + x_{4\,5} = 1 \\
x_{5\,1} + x_{5\,2} + x_{5\,3} + x_{5\,4} = 1 \\
x_{2\,1} + x_{3\,1} + x_{4\,1} + x_{5\,1} = 1 \\
x_{1\,2} + x_{3\,2} + x_{4\,2} + x_{5\,2} = 1 \\
x_{1\,3} + x_{2\,3} + x_{4\,3} + x_{5\,3} = 1 \\
x_{1\,4} + x_{2\,4} + x_{3\,4} + x_{5\,4} = 1 \\
x_{1\,5} + x_{2\,5} + x_{3\,5} + x_{4\,5} = 1 \\
x_{1\,2} + x_{3\,2} + x_{4\,2} + x_{5\,2} \geq 1 \\
x_{1\,3} + x_{2\,3} + x_{4\,3} + x_{5\,3} \geq 1 \\
x_{1\,4} + x_{2\,4} + x_{3\,4} + x_{5\,4} \geq 1 \\
x_{1\,5} + x_{2\,5} + x_{3\,5} + x_{4\,5} \geq 1 \\
x_{2\,1} + x_{3\,1} + x_{4\,1} + x_{5\,1} \geq 1 \\
x_{1\,4} + x_{1\,5} + x_{2\,4} + x_{2\,5} + x_{3\,4} + x_{3\,5} \geq 1 \\
x_{1\,3} + x_{1\,5} + x_{2\,3} + x_{2\,5} + x_{4\,3} + x_{4\,5} \geq 1 \\
x_{1\,3} + x_{1\,4} + x_{2\,3} + x_{2\,4} + x_{5\,3} + x_{5\,4} \geq 1 \\
x_{1\,3} + x_{1\,4} + x_{1\,5} + x_{2\,3} + x_{2\,4} + x_{2\,5} \geq 1 \\
x_{1\,2} + x_{1\,5} + x_{3\,2} + x_{3\,5} + x_{4\,2} + x_{4\,5} \geq 1 \\
x_{1\,2} + x_{1\,4} + x_{3\,2} + x_{3\,4} + x_{5\,2} + x_{5\,4} \geq 1 \\
x_{1\,2} + x_{1\,4} + x_{1\,5} + x_{3\,2} + x_{3\,4} + x_{3\,5} \geq 1 \\
x_{1\,2} + x_{1\,3} + x_{4\,2} + x_{4\,3} + x_{5\,2} + x_{5\,3} \geq 1 \\
x_{1\,2} + x_{1\,3} + x_{1\,5} + x_{4\,2} + x_{4\,3} + x_{4\,5} \geq 1 \\
x_{1\,2} + x_{1\,3} + x_{1\,4} + x_{5\,2} + x_{5\,3} + x_{5\,4} \geq 1 \\
x_{2\,1} + x_{2\,5} + x_{3\,1} + x_{3\,5} + x_{4\,1} + x_{4\,5} \geq 1 \\
x_{2\,1} + x_{2\,4} + x_{3\,1} + x_{3\,4} + x_{5\,1} + x_{5\,4} \geq 1 \\
x_{2\,1} + x_{2\,4} + x_{2\,5} + x_{3\,1} + x_{3\,4} + x_{3\,5} \geq 1 \\
x_{2\,1} + x_{2\,3} + x_{4\,1} + x_{4\,3} + x_{5\,1} + x_{5\,3} \geq 1 \\
x_{2\,1} + x_{2\,3} + x_{2\,5} + x_{4\,1} + x_{4\,3} + x_{4\,5} \geq 1 \\
x_{2\,1} + x_{2\,3} + x_{2\,4} + x_{5\,1} + x_{5\,3} + x_{5\,4} \geq 1 \\
x_{3\,1} + x_{3\,2} + x_{4\,1} + x_{4\,2} + x_{5\,1} + x_{5\,2} \geq 1 \\
x_{3\,1} + x_{3\,2} + x_{3\,5} + x_{4\,1} + x_{4\,2} + x_{4\,5} \geq 1 \\
x_{3\,1} + x_{3\,2} + x_{3\,4} + x_{5\,1} + x_{5\,2} + x_{5\,4} \geq 1 \\
x_{4\,1} + x_{4\,2} + x_{4\,3} + x_{5\,1} + x_{5\,2} + x_{5\,3} \geq 1
\end{cases}$$

Dobbiamo poi considerare il vincolo di interezza $x \in \mathbb{Z}$, e nel caso si prenda il rilassato continuo (che vedremo in questo caso è il problema di assegnamento corrispondente), i $2n$ vincoli in forma $x_{ij} \leq 1$, $x_{ij} \geq 0$.

1.1.2 Valutazioni inferiori e superiori

Vediamo come trovare buone valutazioni inferiori e superiori delle soluzioni v_{TSP} di problemi TSP asimmetrici.

- **Valutazione inferiore:** per fare una valutazione inferiore v_I, si rimuovono i vincoli di connessione, trasformando il problema in un problema di assegnamento (infatti si indica anche $v_I = v_{ASS}$) e concedendo quindi cicli disgiunti. Questo approccio è solitamente molto efficace per i problemi TSP *asimmetrici*, restituendo rapporti $\epsilon = \frac{v_{TSP} - v_{ASS}}{v_{ASS}}$ nell'ordine di poche frazioni di punto percentuale. Nel caso di TSP simmetrici, invece, il rilassamento restituisce errori fino al 60%, e non è quindi particolarmente indicato.

- **Valutazione superiore:** per fare una valutazione superiore v_S, invece, si usa il cosiddetto **algoritmo delle toppe**, o *algoritmo di fusione dei cicli disgiunti* (ma tu vedi gli americani). Questo algoritmo prevede di calcolare il v_I, quindi la soluzione che concede cicli disgiunti, e di selezionare da due di questi un arco ciascuno, (i, j) e (k, l). Si eliminano quindi questi archi, incrociandoli, cioè rimuovendo (i, j) e (k, l) e introducendo (i, l) e (k, j). Avremo variazione di v_{ASS}:

  $$v'_S = v_{ASS} - c_{ij} - c_{kl} + c_{il} + c_{kj}, \quad \Delta v = -c_{ij} - c_{kl} + c_{il} + c_{kj}$$

  Si ripete iterativamente il processo, riducendo di volta in volta il numero di vincoli, finche non si arriva ad un ciclo hamiltoniano. L'algoritmo completo è quindi:

Algoritmo 1 delle toppe

Input: una v_{ASS} soluzione dall'assegnamento ricavato da un TSP
Output: una valutazione inferiore v_I del TSP
ciclo:
Chiama $C = \{C_1, ..., C_p\}$ l'insieme dei cicli (sottocicli) dati da v_{ASS}
Per ogni coppia di cicli (C_h, C_k) valuta l'incremento di costo Δv dovuto alla fusione di C_h e C_k nel modo più conveniente possibile
Effettua la fusione dei cicli C_h e C_k con $\Delta v = \min_{C_i, C_j} \Delta v, \quad i, j \in [1, p], \quad i \neq j$
$p \leftarrow p - 1$
if $p > 1$ **then**
  Vai a ciclo
end if