1 Lezione del 30-10-24

1.1 Problema di copertura

Problema 1.1: di copertura

Supponiamo che l'ASL debba dislocare ambulanze su 5 sedi distribuite sul territorio, in 9 diverse località. La matrice di adiacenza fra sedi e località è la seguente:

	Sede 1	**Sede 2**	**Sede 3**	**Sede 4**	**Sede 5**
Località 1	1	0	1	0	1
Località 2	0	1	0	1	0
Località 3	1	1	0	0	0
Località 4	0	1	1	0	1
Località 5	0	0	0	0	1
Località 6	1	1	1	1	1
Località 7	1	1	0	0	1
Località 8	0	1	1	1	0
Località 9	0	0	1	0	1

Vogliamo capire in quali sedi dovremo dislocare le ambulanze in modo da coprire tutte le località.

Il problema è risolvibile attraverso la ILP. Possiamo assumere che la matrice di adiacenza sia stata ricavata da qualche matrice dei tempi (rappresentante il tempo necessario a raggiungere una località), e quindi tagliata su un tempo massimo di, ad esempio, 20 minuti, con il risultato che le località a meno di 20 minuti di distanza risultano connesse e quelle a più di 20 minuti disconnesse.

Possiamo quindi definire alcune **regole di riduzione** per semplificare la matrice di adiacenza:

1. Le righe di soli zeri $(0, ..., 0)$ possono essere eliminate in quanto rappresentano una soluzione impossibile del problema;
2. Le righe di soli uni $(1, ..., 1)$ possono essere eliminate in quanto possono essere risolte da qualsiasi servizio;
3. Se una riga contiene un solo 1, apro un "servizio" per quella riga (cioè gli dedico una delle colonne), la elimino, e elimino tutte le righe che hanno 1 sulla stessa colonna.
4. Se per due colonne r e k, $r_{ij} \geq k_{kj}$ per ogni riga, allora si diche che r **domina** k, e quindi che k può essere scartata. Questo vale solo se l'apertura è a costo costante su tutte le colonne: in caso contrario potremmo scartare opzioni viabili di ottimizzazione.

Vogliamo capire quali località dovranno essere servite da quale sede in modo da coprirle tutte. Si ricavano quindi i vincoli su ogni riga:

$$\begin{cases} x_1 + x_3 + x_5 \geq 1 \\ x_2 + x_4 \geq 1 \\ \ldots \\ x_3 + x_5 \geq 1 \\ x_i \in \{0, 1\} \end{cases}$$

ergo si ricava un problema di ILP in forma:

$$\begin{cases} \min c^T \cdot x \\ Ax \geq 1 \\ x \in \{0, 1\} \end{cases}$$

dove c rappresenta un vettore costo, nel problema riportato assunto come costante.

Quello che facciamo effettivamente è scegliere fra n sottoinsiemi, scegliendo l'insieme minimo di questi per coprire l'interezza degli elementi (detti anche nodi).

1.1.1 Valutazioni inferiori e superiori

Vediamo quali valutazioni possiamo usare per trovare soluzioni approssimate.

- **Valutazione inferiore:** una valutazione inferiore v_I può essere agevolmente calcolata dal rilassato continuo del problema ILP:

$$\begin{cases} \min c^T \cdot x \\ Ax \geq 1 \\ 0 \leq x \leq 1 \end{cases}$$

- **Valutazione superiore:** si può applicare un algoritmo greedy per il calcolo di una valutazione superiore, assunti costi costanti:

Algoritmo 1 di copertura a costi costanti

Input: un problema di copertura
Output: una valutazione superiore v_S
ciclo:
$r_i \leftarrow$ la somma dei valori su ogni colonna
Rimuovi la colonna con r_i maggiore, scegli quella colonna nella soluzione, e rimuovi le righe servite da quella colonna
Torna a ciclo

Notiamo che r_i rappresenta effettivamente un rendimento, come avevamo visto nei problemi di zaino. Possiamo infatti applicare un'altro algoritmo, più sofisticato, nel caso dei costi variabili:

Algoritmo 2 di Chvatal

Input: un problema di copertura
Output: una valutazione superiore v_S
ciclo:
$r_i \leftarrow$ il rapporto fra il costo di ogni colonna e la somma dei valori su quella colonna
Rimuovi la colonna con r_i maggiore, scegli quella colonna nella soluzione, e rimuovi le righe servite da quella colonna
Torna a ciclo

1.1.2 Problemi di massima copertura

Supponiamo di avere una stima del valore (che possiamo assimilare alla domanda, o al guadagno associato all'apertura di un servizio su un certo nodo) su ogni riga (nell'esempio precedente di ogni località), e di voler prediligere soluzioni che coprono righe con valore maggiore, supponendo di avere un numero limitato di k risorse (in questo caso ambulanze) a disposizione. Prendiamo ad esempio la tabella:

	Abitanti	**Sede 1**	**Sede 2**	**Sede 3**	**Sede 4**	**Sede 5**
Località 1	2000	1	0	1	0	1
Località 2	1000	0	1	0	1	0
Località 3	800	1	1	0	0	0
Località 4	3000	0	1	1	0	1
Località 5	2500	0	0	0	0	1
Località 6	1200	1	1	1	1	1
Località 7	1700	1	1	0	0	1
Località 8	400	0	1	1	1	0
Località 9	150	0	0	1	0	1

Il problema potrebbe essere di assegnare ambulanze alle sedi che offrono copertura della maggior popolazione possibile. Formuleremo allora il problema di ILP:

$$\begin{cases} \max h_1 z_1 + ... + h_9 z_9 \\ x_1 + x_3 + x_5 \geq z_1 \\ x_2 + x_4 \geq z_2 \\ ... \\ x_3 + x_5 \geq z_9 \\ x_1 + x_2 + x_3 + x_4 = k \\ x \in \{0, 1\} \\ z \in \{0, 1\} \end{cases}$$

Questo rappresenta un **problema di massima copertura**. I coefficienti $h_1, ..., h_n$ sono i valori associati ad ogni nodo (qui ad ogni località): la funzione obiettivo risponde meglio all'apertura dei servizi ai nodi con valore alto. I successivi n vincoli rappresentano il fatto che, presa un nodo z, cioè una riga, dobbiamo rendere uguale a 1 almeno una fra le colonne che la servono, cioè dobbiamo aprirgli almeno un servizio. Infine, il vincolo $x_1 + ... + x_n = k$ rappresenta il fatto che vogliamo prendere k servizi, cioè colonne (nel nostro caso distribuire k ambulanze).

Quello che facciamo, effettivamente, è scegliere fra i sottoinsiemi x per massimizzare la copertura sugli elementi z, massimizzando il guadagno ottenuto in base alla scelta degli z (scelti gli z i successivi vincoli determinano quali x dobbiamo aprire per servirli).