Programmazione Matematica a Numeri Interi (Integer Programming - IP)

Problema di ottimizzazione di una funzione obiettivo lineare soggetta al rispetto di un insieme di vincoli lineari in cui tutte o parte delle variabili possono assumere solo valori interi

Mixed IP Problem - MIP

$$\max \underline{c}^T \underline{x} + \underline{h}^T \underline{y}$$
$$A\underline{x} + G\underline{y} \leq \underline{b}$$
$$\underline{x} \in \mathbf{Z}_+^n \quad \underline{y} \in \mathbf{R}_+^p$$

$\underline{x}$ è il vettore delle variabili intere positive
$\underline{y}$ è il vettore delle variabili reali positive

I problemi in cui sono presenti variabili intere sono spesso anche indicati come Problemi Combinatorici.

Alcuni esempi di applicazioni:
- Production Scheduling (assegnazione e sequenziazione di operazioni su macchine)
- Distribuzione di beni
- Facility location (localizzazione di impianti)
- Progettazione di reti (trasporto e comunicazione)
- Pianificazione di investimenti

Formulazione di problemi a numeri interi

I modelli di ottimizzazione a numeri interi vengono introdotti quando si deve ottimizzare l'uso di risorse non divisibili o la scelta tra alternative discrete.

Un tipo di modello molto importante e comune è quello in cui le variabili intere possono assumere solo i valori 0 e 1 (0-1 Programming).

Valori binari sono usati per rappresentare la scelta tra due possibilità o il verificarsi o meno di una condizione:

$$x = \begin{cases} 1 & \text{se l'evento si verifica} \\ 0 & \text{se l'evento non si verifica} \end{cases}$$

Esempi di IP:

- Il problema dello zaino (Knapsack Problem)
- Il problema dell'assegnazione (Matching Problem)
- Il problema del costo fisso (Fixed Charge Problem)
- Il problema del sequenziamento (Sequencing Problem)
- I problemi di copertura, partizione ed impaccamento (Set Covering, Partitioning and Packing Problems)
- Problemi sui grafi (Teoria dei Grafi):
    - Percorso minimo (Shortest Path)
    - Commesso viaggiatore (Traveling Salesman Problem)

Il problema dello zaino
(Knapsack Problem)

Si hanno n possibili progetti da realizzare con un budget massimo b disponibile.
Se un progetto j, j=1,...,n, viene finanziato deve essere investito un capitale a_j.
Dalla realizzazione di un progetto j si ricava un guadagno c_j.
Un progetto non può essere realizzato parzialmente, ma può essere eseguito completamente o non eseguito affatto.

Il problema: stabilire quali progetti realizzare per ottenere il massimo guadagno senza superare il budget disponibile.

Il problema si formula introducendo n variabili binarie:

$$x_j = \begin{cases} 1 & \text{il progetto } j \text{ è finanziato} \\ 0 & \text{il progetto } j \text{ non è finanziato} \end{cases}$$
$$j = 1, \ldots, n$$

Knapsack binario

$$\max \sum_{j=1}^{n} c_j x_j$$
$$\sum_{j=1}^{n} a_j x_j \le b$$
$$x_j \in \{0,1\} = \mathbf{B} \quad \forall j = 1, \ldots, n$$

Problema dello zaino: si deve stabilire quali tra n oggetti portare in uno zaino sapendo che il peso massimo trasportabile è b, che ogni oggetto j pesa a_j e che ad ogni oggetto è associata un'utilità (valore) c_j.

Una variante: la realizzazione di progetti nell'arco di m mesi.

Supponendo che la realizzazione di ciascun progetto richieda un finanziamento nell'arco di m mesi:

- a_{ij} il finanziamento per il progetto j nel mese i, j=1,...,n; i=1,...,m
- b_i budget disponibile nel mese i, i=1,...,m

Knapsack multidimensionale

$$\max \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \forall i = 1, \ldots, m$$

$$x_j \in \{0,1\} = \mathbf{B} \;\; \forall j = 1, \ldots, n$$

Il problema dell'assegnamento (Matching Problem)

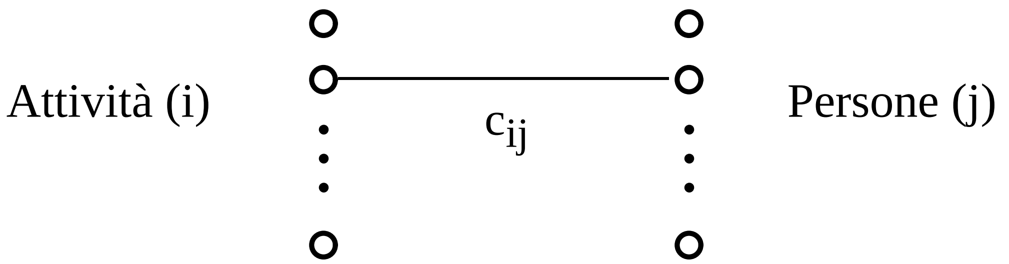
Si devono assegnare m attività ad n persone (sia $n \geq m$).

Una volta stabilito l'assegnamento, ogni persona eseguirà solo l'attività che gli viene assegnata.

Se alla persona j, j=1,...,n, viene assegnata l'attività i, i=1,..., m, il costo pagato è c_{ij}.

Il problema: assegnare tutte le attività alle persone in modo da minimizzare il costo pagato.

Si può rappresentare il problema con un grafo:



Il problema consiste nello stabilire quali archi introdurre tra i nodi "attività" ed i nodi "persona" in modo che su ogni nodo "attività" incida esattamente un arco e su ogni nodo "persona" incida al più un arco.

Il problema si formula introducendo mxn variabili binarie:

$$x_{ij} = \begin{cases} 1 & \text{l'attività } i \text{ è assegnata alla persona } j \\ 0 & \text{l'attività } i \text{ non è assegnata alla persona } j \end{cases}$$

$$i = 1,\dots,m$$

$$j = 1,\dots,n$$

Matching binario

$$\boxed{\begin{aligned} &\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij} \\ &\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i = 1,\dots,m \quad (a) \\ &\sum_{i=1}^{m} x_{ij} \le 1 \quad \forall j = 1,\dots,n \quad (b) \\ &x_{ji} \in \{0,1\} = \mathbf{B} \;\; \forall i = 1,\dots,m \, \forall j = 1,\dots,n \end{aligned}}$$

Vincoli (a): ogni attività e' assegnata ad una persona

Vincoli (b): ogni persona svolge al più una sola attività

Il problema del costo fisso (Fixed Charge Problem)

Si consideri un problema di trasporto tra m produttori ed n consumatori (ad esempio, un trasporto di gas o petrolio).

Ogni produttore è connesso ad ogni consumatore da un canale attraverso cui può avvenire il trasporto (e.g., un gasdotto).

Per avere un flusso trasportato tra un produttore i, i=1,...,m, ed un consumatore j=1,...,n, si deve:

- pagare un costo fisso f_{ij} per l'affitto del canale di trasporto tra i e j;
- pagare c_{ij} per ogni unità trasportata tra i e j.

La disponibilità massima del produttore i è s_i, i=1,...,m.

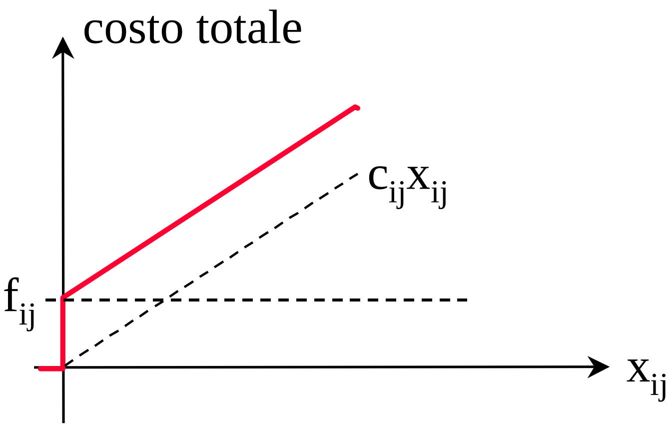
La richiesta del consumatore j che deve essere soddisfatta è r_j, j=1,...,n.

Il problema: stabilire quali canali di trasporto tra i produttori ed i consumatori affittare e quale flusso trasportare su tali canali in modo da minimizzare il costo totale, soddisfacendo la richiesta dei consumatori.

Le variabili:

- $x_{ij} \in \mathbf{R}$, i=1,...,m; j=1,...,n, la quantità trasportata tra i e j
- $y_{ij} \in \mathbf{B}$, i=1,...,m; j=1,...,n, $y_{ij} = \begin{cases} 1 \text{ se si utilizza il canale tra i e j} \\ 0 \text{ se non si utilizza il canale tra i e j} \end{cases}$

La funzione di costo ha una discontinuità nell'origine.
Ad esempio per la variabile x_{ij}



Il problema del trasporto con costo fisso

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij}x_{ij} + f_{ij}y_{ij})$$

$$\sum_{j=1}^{n} x_{ij} \le s_i \quad \forall i = 1,\dots,m \quad (a)$$

$$\sum_{i=1}^{m} x_{ij} = r_j \quad \forall j = 1,\dots,n \quad (b)$$

$$x_{ij} \le My_{ij} \quad \forall i, j \quad (c)$$

$$x_{ij} \in \mathbf{R}_+ \quad y_{ij} \in \mathbf{B} \quad \forall i = 1,\dots,m \; \forall j = 1,\dots,n$$

Vincoli (a) e (b): i vincoli del problema del trasporto.
Vincoli (c): un flusso x_{ij} può esistere solo se si paga il costo fisso per usare il canale tra i e j. M è un coefficiente molto più grande della massima quantità trasportabile tra i e j. Se fosse specificato il flusso massimo, q_{ij}, tra i e j, i vincoli (c) risulterebbero

$$x_{ij} \le q_{ij}y_{ij} \qquad \forall i, j$$

Il problema del sequenziamento
(Sequencing Problem)

Si devono sequenziare n attività (job) indipendenti su una macchina.

La macchina può eseguire solo un job alla volta.

L'esecuzione dei job non può essere interrotta.

Il tempo necessario per l'esecuzione (processing time) del job i, i=1,...,n, è p_i.

Viene pagato un costo che, in generale, è funzione del tempo necessario a completare i job (ad es., il tempo medio di completamento dei job).

<u>Il problema</u>: determinare la sequenza con cui eseguire i job sulla macchina in modo da minimizzare il costo.

Le variabili continue :

- $t_i \in \mathbf{R}_+$, i=1,...,n, l'istante di inizio esecuzione del job i.

Dati due job i e j, si possono verificare due casi:

$$\left.\begin{array}{lll} 1. & \text{i precede j} & \Rightarrow t_j \geq t_i + p_i \\ 2. & \text{j precede i} & \Rightarrow t_i \geq t_j + p_j \end{array}\right\} \underline{\text{Vincoli disgiuntivi}}$$

I due casi sono mutuamente esclusivi, quindi solo un vincolo tra 1. e 2. può essere soddisfatto da una soluzione del problema.

Si introduce un variabile binaria:

$$y_{ij} \in \mathbf{B}, i = 1,\dots,n;\ j = 1,\dots,n \qquad y_{ij} = \begin{cases} 1 & \text{se } i \text{ precede } j \\ 0 & \text{se } j \text{ precede } i \end{cases}$$

quindi per ogni coppia di job, i, j, i vincoli disgiuntivi del problema si esprimono come:

$$t_j \geq t_i + p_i - M(1 - y_{ij}) \quad (a)$$
$$t_i \geq t_j + p_j - My_{ij} \quad (b)$$

dove M è un coefficiente costante positivo scelto molto maggiore di ogni altro coefficiente (in particolare della $\sum_{i=1}^{n} p_i$) del problema.

Se y_{ij}=1:

- il vincolo (a) coincide con il vincolo 1.
- il vincolo (b) è sempre soddisfatto poichè $M >> t_j - t_i + p_j$

Se y_{ij}=0:

- il vincolo (a) è sempre soddisfatto poichè $M >> t_i - t_j + p_i$
- il vincolo (b) coincide con il vincolo 2.

Il vettore delle y_{ij} rappresenta l'ordine di esecuzione dei job.

Problemi di copertura, impaccamento e partizionamento
(Set Covering, Packing and Partitioning Problems)

Una definizione generale dei

Problemi di Ottimizzazione Combinatorica:

Definizione:

Sono dati un insieme finito $N=\{1,...,n\}$ ed un vettore $\underline{c}^T=[c_1,...,c_n]$.

Dato un generico sottinsieme $F \subseteq N$, si definisce c(F) come

$$c(F) = \sum_{j \in F} c_j$$

Sia data una collezione $\mathcal{F}$ di sottinsiemi di N, allora un <u>Problema di Ottimizzazione Combinatorica</u> (POC) è dato da

$$(\text{POC}) \quad \max_{F \in \mathcal{F}} c(F)$$

I problemi di covering, packing e partitioning sono tre tipi di problemi di ottimizzazione combinatorica che differiscono per il modo con cui è definito l'insieme $\mathcal{F}$, ossia per la diversa proprietà che caratterizza gli insiemi F conenuti in $\mathcal{F}$.

Sia dato un insieme finito M={1,...,m}

ed un insieme di sottinsiemi di M, $S = \left\{ M_j \subseteq M : j \in N \right\}$

con N={1,...,n}.

Si definiscono le tre seguenti classi di problemi:

- L'insieme $F \subseteq N$ si dice **Covering di M** se $\bigcup_{j \in F} M_j = M$

  Un problema di **Set Covering** è un POC per cui

  $\mathcal{F}$={F: F è covering di M}

- L'insieme $F \subseteq N$ si dice **Packing di M** se

  $$\forall j,k \in F,\ j \neq k \quad M_j \cap M_k = \emptyset$$

  Un problema di **Set Packing** è un POC per cui

  $\mathcal{F}$={F: F è packing di M}

- L'insieme $F \subseteq N$ si dice **Partitioning di M** se F è sia un covering che un packing di M.

  Un problema di **Set Partitioning** è un POC per cui

  $\mathcal{F}$={F: F è partitioning di M}

Tipicamente i problemi di Set Covering cercano la copertura a costo minimo, per cui i valori c_j rappresentano il costo di M_j;
i problemi di Set Packing cercano l'impaccatura dal valore massimo, per cui i valori di c_j rappresentano il valore di M_j.

Formulazione dei problemi

I POC di Set Covering, Packing e Partitioning possono essere formulati come problemi lineari a numeri binary (0-1 IP).

Si definisce la **Matrice di Incidenza** di S, una matrice A (mxn).
La matrice A ha una riga per ogni elemento di M ed una colonna per ogni sottinsieme $M_j \subseteq M$

$$A = \begin{array}{cc} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} & \begin{array}{c} 1 \\ \vdots \\ m \end{array} \\ \begin{array}{ccc} M_1 & \cdots & M_n \end{array} & \end{array} \qquad a_{ij} = \begin{cases} 1 & \text{se } i \in M_j \\ 0 & \text{se } i \notin M_j \end{cases}$$

Si definisce $\underline{x}$ **Vettore di Incidenza** di F $\underline{x} \in \mathbf{B}^n$ tale che

$$x_j = \begin{cases} 1 & \text{se } j \in F & \text{(viene scelto l'insieme } M_j\text{)} \\ 0 & \text{se } j \notin F & \text{(non viene scelto l'insieme } M_j\text{)} \end{cases}$$

Allora i vincoli delle tre classi di problemi si possono scrivere come:

- Covering: $A\underline{x} \geq \underline{1}$
- Packing: $A\underline{x} \leq \underline{1}$
- Partitioning: $A\underline{x} = \underline{1}$

dove $\underline{1}$ è un vettore a m elementi tale che $\underline{1}^T$=[1,...,1]

Esempi di POC

a) Formazione di gruppi di lavoro.

Sono date m attività ed n persone, ciascuna in grado di svolgere un sottinsieme di tali attività. Ad ogni persona j, j=1,...,n, è associato un costo c_j.
Si vuole costituire un gruppo di lavoro formato da un sottinsieme di persone in maniera che il gruppo sia in grado di svolgere tutte le attività al costo complessivo minimo.

M={1,...,m} l'insieme delle attività
N={1,...,n} l'insieme delle persone
$S = \left\{ M_j \subseteq M : j \in N \right\}$

M_j è il sottinsieme delle attività che la persona j è in grado di svolgere:
ad esempio, se la persona j è in grado di svolgere l'attività i allora $i \in M_j$

La matrice A ha una riga per ciascuna attività ed una colonna per ciascuna persona.

E' un problema di Set Covering. Le variabili costituiscono il vettore di incidenza che indica le persone scelte.

$$\min \underline{c}^T \underline{x}$$
$$A\underline{x} \geq \underline{1}$$
$$\underline{x} \in \mathbf{B}^n$$

b) <u>Localizzazione di impianti (Facility Location)</u>.

Sono date m comunità che devono essere protette da una centrale antiincendio ed n possibili localizzazioni delle centrali.
Il costo pagato per posizionare una centrale nel sito j è c_j.
Una centrale situata in j è in grado di proteggere un certo sottinsieme di comunità.
Si vuole determinare dove localizzare le centrali in modo che tutte le comunità siano protette minimizzando il costo complessivo.

M={1,...,m} l'insieme delle comunità da proteggere
N={1,...,n} l'insieme delle possibili localizzazioni delle centrali
$S = \left\{ M_j \subseteq M : j \in N \right\}$

M_j è il sottinsieme delle comunità che sono protette da una centrale situata in j (ad esempio, perchè è raggiungibile entro un tempo massimo).

La matrice A ha una riga per ciascuna comunità da proteggere ed una colonna per ciascuna possibile localizzazione di una centrale.

E' un problema di Set Covering. Le variabili costituiscono il vettore di incidenza che indica i siti scelti per le centrali.

$$\min \underline{c}^T \underline{x}$$
$$A\underline{x} \geq \underline{1}$$
$$\underline{x} \in \mathbf{B}^n$$

c) Problemi di instradamento (Routing Problem).

Sono dati m clienti che devono essere serviti da una azienda di trasporti ed n possibili rotte effettuate dai mezzi dell'azienda..
Il costo pagato per seguire una rotta j è c_j.
Una rotta j è in grado di servire un certo sottinsieme di clienti.
Si vuole determinare quali rotte effettuare in modo che tutte i clienti siano serviti con il minimo costo complessivo.

$M=\{1,...,m\}$ l'insieme dei clienti da servire
$N=\{1,...,n\}$ l'insieme delle possibili rotte
$S = \left\{ M_j \subseteq M: j \in N \right\}$

M_j è il sottinsieme dei clienti che sono serviti da un trasporto che segua la rotta j.

La matrice A ha una riga per ciascun cliente da servire ed una colonna per ciascuna possibile rotta da far eseguire ai mezzi di trasporto.

E' un problema di Set Covering. Le variabili costituiscono il vettore di incidenza che indica quali rotte sono state scelte.

$$\min \underline{c}^T \underline{x}$$
$$A\underline{x} \geq \underline{1}$$
$$\underline{x} \in \mathbf{B}^n$$

d) Formazione dei collegi elettorali (Political Districting).

Devono essere suddivisi m comuni in un insieme di distretti elettorali.
Sono specificate a priori n possibili suddivisioni in maniera che rispondano a determinati requisiti.
Se si sceglie di costituire il distretto j si paga un costo c_j.
Ogni comune deve essere incluso esattamente in un solo distretto elettorale.
Si vuole determinare come formare i distretti elettorali in maniera da minimizzare il costo complessivo.

$M=\{1,...,m\}$ l'insieme dei comuni
$N=\{1,...,n\}$ l'insieme dei possibili distretti
$S=\left\{M_j \subseteq M: j \in N\right\}$

M_j è il sottinsieme di comuni che sono inclusi nel distretto elettorale j.

La matrice A ha una riga per ciascun comune ed una colonna per ciascun possibile distretto elettorale.

E' un problema di Set Partitioning. Le variabili costituiscono il vettore di incidenza che indica quali sono i distretti che si vogliono formare.

$$\min \underline{c}^T \underline{x}$$
$$A\underline{x} = \underline{1}$$
$$\underline{x} \in \mathbf{B}^n$$

Metodi di soluzione dei problemi di IP

Consideriamo un problema (IP)

$$
\begin{aligned}
\text{(IP)} \quad \max\ x_0 &= \underline{c}^T \underline{x} \\
A\underline{x} &= \underline{b} \\
\underline{x} &\in \mathbf{Z}_+^n
\end{aligned}
$$

Il poliedro P associato ai vincoli di (IP) contiene tutti e soli i punti interi che sono soluzioni di (IP):

$$P = \left\{ \underline{x} \in \mathbf{Z}^n : A\underline{x} = \underline{b}, \underline{x} \geq \underline{0} \right\}$$

Il **rilassamento lineare** (RL) di (IP) corrisponde a rimuovere da (IP) il vincolo di integrità:

$$
\begin{aligned}
\text{(RL)} \quad \max\ x_0 &= \underline{c}^T \underline{x} \\
A\underline{x} &= \underline{b} \\
\underline{x} &\in \mathbf{R}_+^n
\end{aligned}
$$

Il poliedro P’ associato al (RL) è

$$P' = \left\{ \underline{x} \in \mathbf{R}^n : A\underline{x} = \underline{b}, \underline{x} \geq \underline{0} \right\}$$

$$P = P' \cap \mathbf{Z}^n$$

quindi P’ contiene anche tutte le soluzioni ammissibili per (IP).

Sia $\underline{x}^*$ la soluzione ottima per (IP) (se esiste) e sia $\overline{\underline{x}}$ la soluzione ottima per (RL).
E' evidente che $\underline{c}^T \overline{\underline{x}} \geq \underline{c}^T \underline{x}^*$.
Quindi se $\overline{\underline{x}} \in P$, la soluzione ottima di (RL) è anche la soluzione ottima di (IP).
In generale, tuttavia, $\overline{\underline{x}} \notin P$.

Esistono tre classi di metodi di soluzione dei problemi (IP):

1) Metodi basati su una enumerazione implicita delle soluzioni
(*Branch and Bound Methods*)

2) Metodi basati sull'uso di "piani di taglio"
(*Cutting Planes Methods*)

3) Metodi specifici per particolari classi di problemi
(ad esempio, per il set covering, knapsack, ecc.)

Condizioni per l'interezza della soluzione di (RL)

Consideriamo un problema (IP) per cui A e $\underline{b}$ sono formate da coefficienti interi.

Se (RL) ha una soluzione ottima finita allora ha una soluzione ottima di base. Data una matrice di base B, la soluzione corrispondente è $\underline{x}_B = B^{-1}\underline{b}$

Allora se tutte le soluzioni ammissibili di base di (RL) fossero intere, anche la soluzione ottima di (RL) sarebbe intera.

Poichè $\underline{b}$ è intero, data B, la soluzione di base è intera se B^{-1} è una matrice a numeri interi.

Definizione:
B matrice mxm a valori interi si dice <u>unimodulare</u> se det(B)={-1,1}.

Definizione:
A matrice mxn è detta <u>totalmente unimodulare</u> se ogni sua sottomatrice quadrata non singolare è unimodulare.

1. Teorema (sufficienza):
Se A è totalmente unimodulare ogni soluzione ammissibile di base di (RL) è intera.

La condizione in generale non è necessaria. Tuttavia, nel caso di vincoli di disuguaglianza vale il seguente teorema:

2. Teorema (Hoffman, Kruskal):
Sia A matrice a coefficienti interi.
Le tre seguenti affermazioni sono equivalenti:

1) A è totalmente unimodulare;
2) i punti estremi (se esistono) di $P = \left\{ \underline{x} \in \mathbf{R}^n : A\underline{x} \le \underline{b}, \underline{x} \ge \underline{0} \right\}$ sono interi per ogni vettore $\underline{b}$ intero;
3) ogni sottomatice quadrata non singolare di A ha inversa a coefficienti interi.

Verificare la condizione di totale unimodularità della matrice A è molto complesso.
Nel 1980 questo problema è stato risolto da P.D. Seymour per mezzo di un algoritmo polinomiale.

Una condizione necessaria perchè $A=[a_{ij}]$ sia totalmente unimodulare è che $a_{ij}=\{0,1,-1\}$.

La condizione tuttavia non è sufficiente; ad esempio

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \Rightarrow \det(A) = -2$$

Sono state definite molte condizioni sufficienti per matrici con coefficienti in $\{0,1,-1\}$. Una condizione sufficiente importante poichè si applica alle matrici di incidenza dei grafi è data dal seguente teorema:

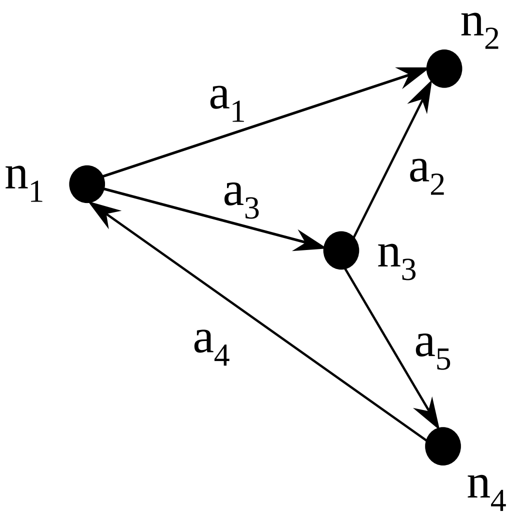
3. Teorema

Sia $A=[a_{ij}]$ matrice tale che $a_{ij}=\{0,1,-1\}$, $i=1,...,m$, $j=1,...,n$.

A è totalmente unimodulate se:

1) ogni colonna di A contiene solo due elementi non nulli;
2) gli indici delle righe di A possono essere partizionati in due insiemi Q_1 e Q_2 tali che:
   - 2.1) se una colonna j contiene due elementi non nulli a_{ij} e a_{hj} che hanno lo stesso segno, allora $i \in Q_1$ e $h \in Q_2$
   - 2.2) se una colonna j contiene due elementi non nulli a_{ij} e a_{hj} che hanno segno opposto, allora i ed h appartengono entrambe a Q_1 oppure a Q_2

Un esempio: Un grafo orientato G



la matrice d'incidenza

$$A = \begin{array}{c} \begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array} \\ \begin{bmatrix} -1 & 0 & -1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ n_1 \\ n_2 \\ n_3 \\ n_4 \end{array}$$

Applicando il teorema:

- a_1 contiene due elementi con segno opposto, $Q_1=\{1, 2\}$, $Q_2=\emptyset$
- a_2 contiene due elementi con segno opposto, $Q_1=\{1, 2, 3\}$, $Q_2=\emptyset$
- a_3 contiene due elementi con segno opposto, $Q_1=\{1, 2, 3\}$, $Q_2=\emptyset$
- a_4 contiene due elementi con segno opposto, $Q_1=\{1, 2, 3, 4\}$, $Q_2=\emptyset$
- a_5 contiene due elementi con segno opposto, $Q_1=\{1, 2, 3, 4\}$, $Q_2=\emptyset$

Il teorema è sempre soddisfatto dalle matrici d'incidenza dei grafi orientati, quindi tali matrici sono sempre totalmente unimodulari.

In generale se il grafo non è orientato la matrice d'incidenza associata può non soddisfare il teorema.

4. **Teorema**

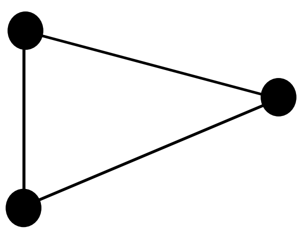
Per tutte le matrici che soddisfano la condizione (1) del teorema 3., la condizione (2) del teorema 3. è necessaria.

Una matrice d'incidenza di un grafo non orientato è totalmente unimodulare se e solo se soddisfa il teorema 3.

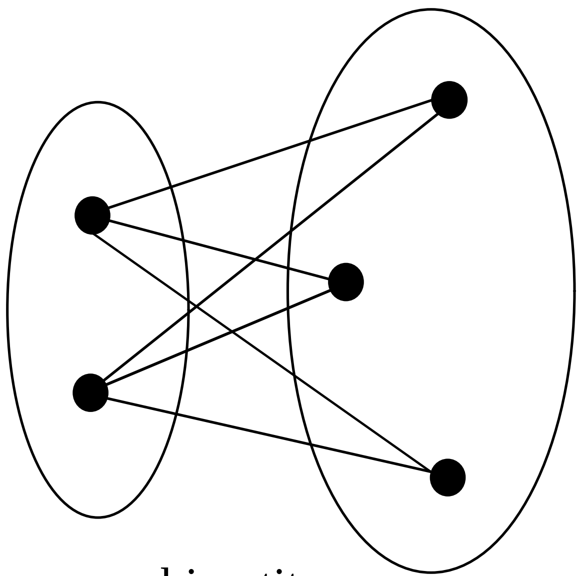
5. Teorema

La matrice d'incidenza di un grafo non orientato è totalmente unimodulare se e solo se il grafo è bipartito.

Un esempio:



non bipartito



bipartito