

HOW TO OTTIMIZZAZIONE

MODULO 1	3
INTRODUZIONE ALLA MODELLIZZAZIONE	3
ESEMPIO	3
TIPI DI PROBLEMA	3
ALGORITMI	4
RILASSAMENTI	4
PROGRAMMAZIONE LINEARE	5
RELAZIONI LOGICHE	5
VINCOLI DI ASSEGNAMENTO	6
SEMI ASSEGNAMENTO	6
INSIEMI AMMISSIBILI	6
ASSEGNAMENTO	6
SELEZIONE DI SOTTOINSIEMI	6
VARIABILI E VALORI DISCRETI	7
MINIMA QUANTITÀ POSITIVA PREFISSATA	7
FUNZIONE CON CARICO FISSO	8
RAPPRESENTARE IL VALORE ASSOLUTO	8
FUNZIONI LINEARI A TRATTI	9
MODULO 2	10
RETI	10
VINCOLI	11
PROBLEMA DEL FLUSSO DI COSTO MINIMO (MCF)	11
PROBLEMA DEL FLUSSO MASSIMO (MF)	13
VINCOLI	13
TAGLI	13
GRAFI RESIDUI	15
CAMMINI AUMENTANTI	15
FORD FULKERSON (teoremi e complessità)	15
EDMONDS-KARP	17
GOLDBERG - TARJAN	18
PROBLEMA DEL FLUSSO MINIMO	19
ALGORITMI AUSILIARI	21
CAMMINI MINIMI SUCCESSIVI	21
COSTRUIRE UN FLUSSO AMMISSIBILE	22
CANCELLAZIONE DEI CICLI	22

PROBLEMI DI ACCOPPIAMENTO	23
ACCOPPIAMENTO DI MASSIMA CARDINALITA'	23
MODULO 3	25
Nozioni preliminari	25
Vertici	25
Soluzioni di Base	26
Inviluppi Convessi	26
Coni Convessi	26
Teorema di Motzkin	27
Dualità	28
Primale e Duale	28
Teorema Debole di Dualità	29
Corollario	29
Dimostrazione	29
Corollario	29
Dimostrazione	29
Direzioni Ammissibili	30
Direzioni di Crescita	30
L'ALGORITMO DEL SIMPLESSO	30
Correttezza del simplesso	31
Complessità del simplesso	32

MODULO 1

INTRODUZIONE ALLA MODELLIZZAZIONE

Dato un problema di ottimizzazione P si inizia fornendo l'insieme F_p delle sue soluzioni ammissibili.

- Di solito si utilizza un insieme G di cui F_p è sottoinsieme (es. \mathbb{N} o \mathbb{R}) e si utilizzano dei vincoli che devono essere rispettati dagli elementi g di G per far parte di F_p .
- Gli elementi di G che non fanno parte di F_p si dicono soluzioni non ammissibili

Successivamente si specifica una funzione obiettivo nella forma $c_p: F_p \rightarrow \mathbb{R}$ che serve a misurare il costo o il beneficio delle soluzioni ammissibili. I problemi possono essere:

- Di massimo: in questo caso dobbiamo determinare $Z_p = \max \{c_p(g) \mid g \in F_p\}$
- Di minimo: in questo caso dobbiamo determinare $Z_p = \min \{c_p(g) \mid g \in F_p\}$
 - Z_p è detto valore ottimo per $P \rightarrow$ un $g^* \in Z_p \mid c_p(g^*)$ è detto soluzione ottima
 - Dato un valore ottimo possiamo ricostruire una soluzione ottima ma non vale il contrario
- Ad ogni problema di massimo corrisponde un problema di minimo P' tale che

$$c_p'(g) = -c_p(g) \rightarrow Z_p = -\min \{c_p'(g) \mid g \in F_p = F_{p'}\}$$

ESEMPIO

- $G = \mathbb{R}$;
- $F_p = \{x \in \mathbb{R} \mid 5x^2 - 6x + 1 = 0\}$; //Soluzioni ammissibili
- $c_p: \mathbb{R} \rightarrow \mathbb{R}; c_p(g) = g^2$; //Funzione obiettivo
- $Z_p = \max \{x^2 \mid 5x^2 - 6x + 1 = 0\}$; //Definizione di Z_p
- Soluzione ottima $g^* = (1, \frac{1}{5})$;
- Valore ottimo $Z_p = \max (1, \frac{1}{5}) = 1$

TIPI DI PROBLEMA

- Problema vuoto:
 - $F_p = \emptyset \rightarrow$ per convenzione $Z_p = \infty$
 - Potrebbe non essere semplice rilevare che un problema è vuoto
- Problema illimitato:
 - Non esiste un valore ottimo per il problema
 - Nel caso di problema di massimo, $\forall x \in \mathbb{R}$ esiste $g \in F_p$ con $c_p(g) \geq x$.
In tal caso $Z_p = +\infty$.
 - Si procede allo stesso modo nel caso di problema di minimo
 - Esempio di problema illimitato:
 - $F_p = \langle x, y \rangle$ tali che $x, y > 0$; problema: massimizzare $x+y$
- Valore ottimo finito, ma non soluzione ottima finita:
 - Z_p esiste ed è finito ma $c_p(g) \neq Z_p \forall g$
 - Esempio: $\inf \{x \mid x > 0\} \rightarrow$ il valore ottimo sarebbe 0 ma non è incluso nell'insieme
- Valore ottimo finito e soluzione ottima finita

Problema di decisione: determinare una qualunque $g \in F_p$ o concludere che $F_p = \emptyset$ (vuoto)

Ad ogni problema di decisione è associato il relativo problema di certificato per G che consiste nel dire se $g \in F_p$ data $g \in G$

Dato un problema di decisione, è sempre possibile vedere quest'ultimo come problema di ottimizzazione considerando F_p costante.

Vale anche il contrario: dato un problema P di ottimizzazione si può considerare R decisionale in modo tale che $F_R = \{g \in F_p \mid c_p(g) = Z_p\}$ da questo deduciamo che per formulare R decisionale dobbiamo conoscere il valore ottimo di P .

Dato $k \in \mathbb{R}$, si può anche considerare R_k decisionale con $F_{R_k} = \{g \in F_p \mid c_p(g) \leq k\}$.

Scelgo k con valori sempre più piccoli risolvendo i vari R_k per avvicinarmi al valore ottimo \rightarrow quando k è troppo piccolo ho un problema vuoto \rightarrow alla fine trovo Z_p

(Ciò vale se P è un problema di minimo, altrimenti si valuta il duale)

ALGORITMI

Esistono due tipi di algoritmo per la risoluzione di un problema P :

- Algoritmi esatti: presa in input un'istanza di P , forniscono in output una soluzione ottima g di P (se esiste) \rightarrow per molti problemi hanno complessità troppo alta
- Gli algoritmi euristici: determinano una qualsiasi soluzione ammissibile e quindi calcolano implicitamente un'approssimazione superiore (se il problema è di minimo) o un'approssimazione inferiore (se il problema è di massimo) del valore ottimo.
 - potrebbero concludere che non esiste soluzione ammissibile anche se $F_p \neq \emptyset$
 - Dati P e $g \in F_p$ esistono due tipologie di errori causati dall'approssimazione:
 - Errore assoluto di g : $\varepsilon_p = c_p(g) - Z_p$
 - Errore relativo di g : $R_p(g) = \frac{\varepsilon_p(g)}{|Z_p|} = \frac{c_p(g) - Z_p}{|Z_p|}$
 - Una soluzione si dice ε -ottima se $R_p(g) < \varepsilon$
 - Un algoritmo euristico si dice ε -approssimato se produce soluzioni ε -ottime

RILASSAMENTI

In alcuni casi è utile risolvere un'approssimazione del problema di partenza.

Un rilassamento di P è un qualunque problema di P^* definito come: $\min\{c_{p^*}(g) \mid g \in F_{p^*}\}$

dove $F_{p^*} \supseteq F_p$ e $\forall g \in F_p \cdot c_{p^*}(g) \leq c_p(g)$.

Ovviamente in caso di problema di massimo si farà riferimento al problema duale.

Il valore Z_{p^*} è minore di Z_p .

Possiamo osservare che molte volte i rilassamenti ammettono soluzioni di complessità inferiore e se la soluzione ottima g^* di P^* soddisfa $g^* \in F_p$ e $c_{p^*}(g^*) = c_p(g^*)$ allora:

$c_{p^*}(g^*) = Z_{p^*} \leq Z_p \leq c_p(g^*) = c_{p^*}(g^*)$ ovvero: Il valore ottimo del problema con rilassamenti sarà minore o uguale a quello del problema di partenza che a sua volta sarà minore o uguale alla funzione obiettivo del problema di partenza applicata alla soluzione ottima del problema con rilassamenti.

PROGRAMMAZIONE LINEARE

Una categoria di problemi dello stesso tipo si dice **modello**.

Un modello particolarmente espressivo che ammette anche soluzioni particolarmente efficienti è la **programmazione lineare (PL)**.

Un problema di programmazione lineare è un problema di ottimizzazione definito dando:

- Un numero finito $n \in \mathbb{N}$ di variabili reali $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ ($G = \mathbb{R}^n$);
- Una funzione obiettivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$ nella forma $f(x) = cx$
 - (funzione lineare tipo $c_1x_1 \dots c_nx_n$)
- Un insieme di m vincoli lineari, tutti in una delle forme seguenti:
 - $ax = b$
 - $ax \leq b$
 - $ax \geq b$

dove a e b sono **parametri** e $a \in \mathbb{R}^n$, $b \in \mathbb{R}$.

Spesso è utile considerare che $x \in \mathbb{N}^n$, ovvero che le soluzioni ammissibili siano composte da numeri naturali \rightarrow in questo caso si parla di **programmazione lineare intera (PLI)**.

Un problema di PL può sempre essere espresso nella forma seguente: $\max\{cx \mid Ax \leq b\}$ dove $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$; cx è la funzione obiettivo; il numero di righe di A è il numero di vincoli; il numero di colonne di A è il numero di elementi di x ; b sono i termini noti.

In particolare:

- Se il problema P è un problema di minimo, basta considerare $f(x) = (-c)x$.
- Ogni vincolo $ax = b$ diventa la coppia di vincoli $ax \leq b$ e $ax \geq b$.
- Ogni vincolo $ax \geq b$ è equivalente a $(-a)x \leq (-b)$.

Mentre in PL le variabili rappresentano delle quantità, in PLI possono essere:

- Quantitative, ovvero rappresentano quantità.
- Logiche, ovvero rappresentano valori binari, booleani, in questo caso deve valere che
 - $x \in \mathbb{N}$; $0 \leq x$; $x \leq 1$, ovvero $0 \leq x \leq 1$

Le variabili logiche possono essere utilizzate per modellare l'assegnamento di una risorsa ad un task o il fatto che una certa attività si debba eseguire oppure no.

RELAZIONI LOGICHE

Negazione ($y = \neg x$) $x = 1 - y.$	Implicazione ($z = (x \rightarrow y)$) $x + z \geq 1;$ $z \geq y;$ $x + z \leq 1 + y.$
Congiunzione ($z = (x \wedge y)$) $z \leq x;$ $z \leq y;$ $z \geq x + y - 1.$	Disgiunzione ($z = (x \vee y)$) $z \geq x;$ $z \geq y;$ $z \leq x + y.$

VINCOLI DI ASSEGNAMENTO

Si parte da un insieme $N = \{1, \dots, n\}$ di oggetti e un insieme $V = \{1, \dots, m\}$ di luoghi.

Luoghi e oggetti non devono necessariamente essere "fisici".

L'utilità di questi vincoli è quella di rappresentare le varie condizioni in cui assegnare oggetti a luoghi (ad esempio una variabile x_{ij} prende valori in $\{0; 1\}$ e modella il fatto che l' i -esimo oggetto è stato assegnato al j -esimo luogo).

SEMI ASSEGNAMENTO

Ogni oggetto è assegnato ad un luogo: $\sum_{j=1}^m x_{ij} = 1$ con $1 \leq i \leq n$

INSIEMI AMMISSIBILI

Talvolta, ogni oggetto $i \in \{1, \dots, n\}$ può essere assegnato ad uno specifico insieme $B(i) \subseteq V$ di luoghi, in tal caso, x_{ij} esiste solo se $i \in B(i)$ e il vincolo precedente diventa $\sum_{j \in B(i)} x_{ij} = 1$

ASSEGNAMENTO

Ogni oggetto è assegnato ad un luogo e ad ogni luogo è assegnato un oggetto:

$$\sum_{j=1}^m x_{ij} = 1 \text{ con } 1 \leq i \leq n \text{ e } \sum_{i=1}^n x_{ij} = 1 \text{ } 1 \leq j \leq m$$

Possono essere un modo per imporre che gli n lavori siano eseguiti in un certo ordine.

La variabile x_{ij} indicherà quindi se l' i -esimo lavoro è effettuato come j -esimo (se vale 1) o meno (se vale 0).

SELEZIONE DI SOTTOINSIEMI

- Sia $N = \{1, \dots, n\}$ un insieme finito di elementi e sia poi $F = \{F_1, \dots, F_m\}$ una famiglia di suoi sottoinsiemi, dove $F_i \subseteq N$.
- Ad ogni F_j (con $1 \leq j \leq m$) associamo un costo c_j .
- Vogliamo determinare $D \subseteq F$ di costo minimo, tra tutti i sottoinsiemi di F che soddisfano certi vincoli. (Non $D = \emptyset$)
- Tale situazione può essere rappresentata con una matrice $A = (a_{ij}) \in \{0, 1\}^{n \times m}$ dove

$$a_{ij} = \begin{cases} 1 & \text{se } i \in F_j; \\ 0 & \text{altrimenti.} \end{cases}$$

- Il vettore delle variabili avrà la forma $x = (x_1, \dots, x_m)$ dove

$$x_j = \begin{cases} 1 & \text{se } F_j \in D; \\ 0 & \text{altrimenti.} \end{cases}$$

- La funzione obiettivo, da minimizzare, sarà sempre $m \sum_{j=1}^m c_j x_j$. ($x \in \{0, 1\}$)
- I vincoli dipendono invece dal problema:
 - **Problema di copertura:** ognuno degli elementi di N sta in almeno in uno degli elementi di D, quindi la funzione da minimizzare andrà posta ≥ 1
 - es: azienda deve assumere programmatori per i linguaggi che servono: anche se due conoscono lo stesso linguaggio l'importante è che siano tutti coperti (minimizzando il costo)
 - **Problema di partizione:** ognuno degli elementi di N sta in esattamente uno degli elementi di D, quindi la funzione da minimizzare andrà posta = 1
 - es: azienda deve appaltare alcuni progetti a vari esecutori. Ogni esecutore scelto dovrà completare i progetti assegnati (minimizzando il costo) e nessun progetto può essere assegnato a più esecutori
 - **Problema di riempimento:** ognuno degli elementi di N sta in al più uno degli elementi di D, quindi la funzione da minimizzare andrà posta ≤ 1
 - es: costruisco un computer con vari pezzi in modo da massimizzare il ricavo \rightarrow anche se avanzano dei pezzi fa lo stesso ma un pezzo non può stare in due computer

VARIABILI E VALORI DISCRETI

- Spesso le variabili in gioco sono vincolate a prendere il loro valore da un insieme che:
 - Non è semplicemente $\{0, 1\}$.
 - Non è \mathbb{N} .
 - Non è un intervallo.
- Per esempio, potremmo essere interessati a vincolare x a stare nell'insieme $\{v_1, \dots, v_n\}$ dove i v_i sono valori reali distinti.
- In tal caso, procederemo introducendo n variabili y_1, \dots, y_n vincolate come segue:

$$y_i \in \{0, 1\} \quad \sum_{j=1}^n y_j = 1 \quad x = \sum_{j=1}^n v_j y_j$$

MINIMA QUANTITÀ POSITIVA PREFISSATA

- Quando una variabile x rappresenta un certo livello di produzione, capita spesso che il valore di tale variabile debba viaggiare in un insieme: $\{0\} \cup [l, u]$.
dove 0 rappresenta l'assenza di produzione, mentre l'intervallo $[l, u]$ rappresenta i possibili livelli di produzione quando il meccanismo è attivo.
- Per modellare tutto questo:
 - Introduciamo una variabile logica $y \in \{0, 1\}$ che indica la presenza o meno di produzione.
 - I vincoli saranno poi

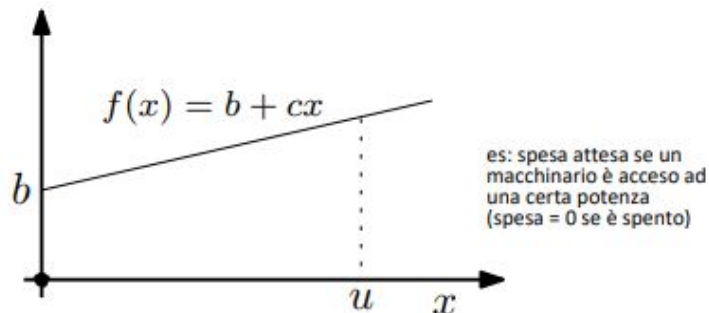
$$ly \leq x \quad x \leq uy.$$
 - Correttamente, se $y = 0$, allora $x = 0$. Altrimenti, $l \leq x \leq u$.

FUNZIONE CON CARICO FISSO

- Si supponga di lavorare con la seguente funzione con carico fisso (dove $b, c > 0$), da minimizzare:

$$x_i = \begin{cases} 0 & \text{se } x = 0; \\ b + cx & \text{se } x \in (0, u]. \end{cases} \quad (\text{Intervallo semichiuso})$$

- La situazione è dunque la seguente:



- Introduciamo ora una variabile logica y , che rappresenta, intuitivamente, la presenza di produzione.

- Occorreranno i seguenti due vincoli:

$$0 \leq x \leq yu \rightarrow 0 \leq x \leq yu$$

- La funzione sarà rappresentata tramite una nuova funzione

$$g(x, y) = by + cx.$$

- Abbiamo infatti che

$$g(0, 0) = 0; \quad g(x, 1) = b + cx.$$

- Si noti che se $y = 0$, allora $x = 0$, ma che non vale il viceversa. In altre parole i due valori $g(0, 1)$ e $g(0, 0)$ sono diversi, ma corrispondono entrambi a soluzioni ammissibili (ovvero $(0, 1)$ e $(0, 0)$). La funzione va però minimizzata e quindi si sceglie correttamente $g(0, 0)$.

RAPPRESENTARE IL VALORE ASSOLUTO

- **Nei Vincoli**

- $g(x) \leq b \quad -g(x) \leq b$ (non è sempre possibile)

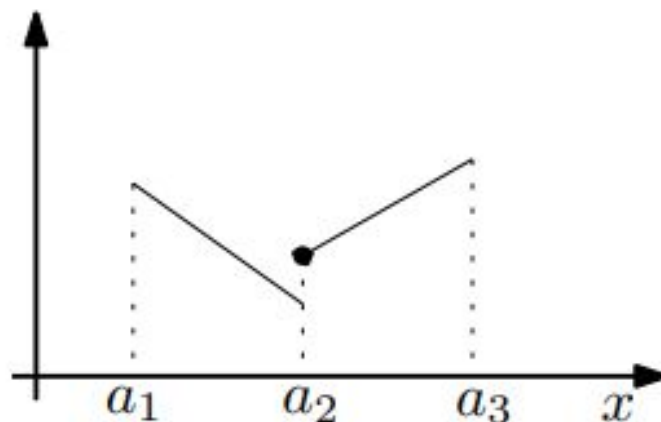
- **Nella Funzione Obiettivo**

- $\max \{ f(x) \mid x \in X \} \quad \max \{ -f(x) \mid x \in X \}$ (confronto rispettivi valori ottimi)

FUNZIONI LINEARI A TRATTI

- Un problema molto interessante è proprio quello di rappresentare funzioni lineari a tratti, eventualmente con l'ausilio di variabili logiche.
- Supponiamo di essere nella situazione seguente:

$$f(x) = \begin{cases} b_1 + c_1x & \text{se } x \in [a_1, a_2]; \\ b_2 + c_2x & \text{se } x \in (a_2, a_3]. \end{cases}$$



- In analogia con quanto fatto per il carico fisso, introduciamo due variabili logiche ausiliarie y_1, y_2 con il significato seguente

$$y_1 = \begin{cases} 1 & \text{se } x \in [a_1, a_2]; \\ 0 & \text{altrimenti.} \end{cases} \quad y_2 = \begin{cases} 1 & \text{se } x \in (a_2, a_3]; \\ 0 & \text{altrimenti.} \end{cases}$$

•

Inoltre, introduciamo due altre variabili quantitative ausiliarie z_1 e z_2 tali che:

$$z_1 = \begin{cases} x - a_1 & \text{se } x \in [a_1, a_2]; \\ 0 & \text{altrimenti.} \end{cases} \quad z_2 = \begin{cases} x - a_2 & \text{se } x \in (a_2, a_3]; \\ 0 & \text{altrimenti.} \end{cases}$$

- Tutto questo è catturabile tramite i vincoli seguenti:

$$\begin{aligned} 0 \leq z_1 \leq (a_2 - a_1)y_1 & & y_1 + y_2 = 1 \\ 0 \leq z_2 \leq (a_3 - a_2)y_2 & & y_1, y_2 \in \{0, 1\} \end{aligned}$$

A questo punto rappresentiamo la funzione f attraverso $g : \mathbb{R}^4 \rightarrow \mathbb{R}$, definita come segue:

$$\begin{aligned} g(z_1, z_2, y_1, y_2) &= b_1 y_1 + c_1 (a_1 y_1 + z_1) + b_2 y_2 + c_2 (a_2 y_2 + z_2) \\ &= (b_1 + c_1 a_1) y_1 + c_1 z_1 + (b_2 + c_2 a_2) y_2 + c_2 z_2 \end{aligned}$$

- Il valore di f in ogni punto $x \in [a_1, a_3]$ è rappresentato univocamente da una quadrupla di valori (z_1, z_2, y_1, y_2) .
 - L'unica eccezione è $x = a_2$, che corrisponde alle due quadruple seguenti:
 $(a_2 - a_1, 0, 1, 0)$ $(0, 0, 0, 1)$ dei quali solo il primo è accettabile
 - Se supponiamo il problema sia un problema di minimo, allora possiamo considerare il problema come benigno, visto che nel punto di discontinuità f cresce.
 - Tutto questo può essere generalizzato al caso di funzioni lineari a $n > 2$ tratti.
-

MODULO 2

RETI

Con il termine rete indichiamo un grafo $G = (N, A)$, di solito diretto, ai cui archi siano associati dei pesi.

Dentro gli archi (insieme A) fluiscono oggetti rappresentati da grandezze continue $\in \mathbb{R}$ o discrete (variabili logiche), mentre i nodi (insieme N) possono essere visti come punti di ingresso, **passaggio** o uscita dalla rete.

Ad ogni nodo $i \in N$ è associato un parametro reale b_i , detto sbilanciamento, che può essere:

- Positivo: il nodo i è un nodo di **uscita** dalla rete e viene detto destinazione, pozzo, oppure nodo di output.
 - b_i è detto domanda di i .
- Negativo: Il nodo i è un nodo di entrata nella rete e viene detto origine, sorgente, oppure nodo di input.
 - $-b_i$ è detto offerta di i .
- Nullo: il nodo i è detto nodo di trasferimento

Ad ogni arco $(i; j) \in A$ sono associati:

- Un costo c_{ij} che indica quanto costa, per un'unità di bene, attraversare il canale.
- Una capacità inferiore l_{ij} , ossia un limite inferiore alla quantità di beni che possono attraversare il canale.
- Una capacità superiore u_{ij} , ossia un limite superiore alla quantità di beni che possono attraversare il canale.

Una **soluzione** per un problema di flusso è un assegnamento di valori reali agli archi di una rete $G = (N, A)$, formalizzato tramite una serie di variabili x_{ij} corrispondenti agli archi $(i,j) \in A$.

Il **costo** di un flusso è il costo complessivo di tutti i flussi nella rete: $\sum_{(i,j) \in A} c_{ij} * x_{ij}$

VINCOLI

Domanda e offerta globale si equivalgono: la somma degli sbilanciamenti positivi è uguale a quella degli sbilanciamenti negativi cambiata di segno se e solo se lo sbilanciamento complessivo è nullo, ovvero:

$$\sum_{i \in D} b_i = - \sum_{i \in O} b_i \Leftrightarrow \sum_{i \in N} b_i = 0$$

Dove $D = \{i \in N \mid b_i > 0\}$ e $O = \{i \in N \mid b_i < 0\}$

Il flusso si conserva: la somma dei flussi entranti nel grafo meno la somma dei flussi uscenti è uguale allo sbilanciamento complessivo, ovvero:

$$\sum_{(j,i) \in BS(i)} x_{ji} - \sum_{(i,j) \in FS(i)} x_{ij} = b_i \quad i \in N,$$

dove

\downarrow ENTRANTI \downarrow USCENTI

$BS(i) = \{(k, i) \mid (k, i) \in A\}; \rightarrow$ STELLA ENTRANTE
 $FS(i) = \{(i, k) \mid (i, k) \in A\}; \rightarrow$ STELLA USCENTE

Il flusso deve essere ammissibile: il flusso deve essere compreso tra la capacità minima e quella massima, ovvero: $l_{ij} \leq x_{ij} \leq u_{ij}$

PROBLEMA DEL FLUSSO DI COSTO MINIMO (MCF)

In questo problema dobbiamo minimizzare il costo del flusso (che è la funzione obiettivo), considerando le capacità inferiori nulle.

$$\begin{aligned} \min cx \\ 0 \leq x \leq u \quad Ex = b \end{aligned}$$

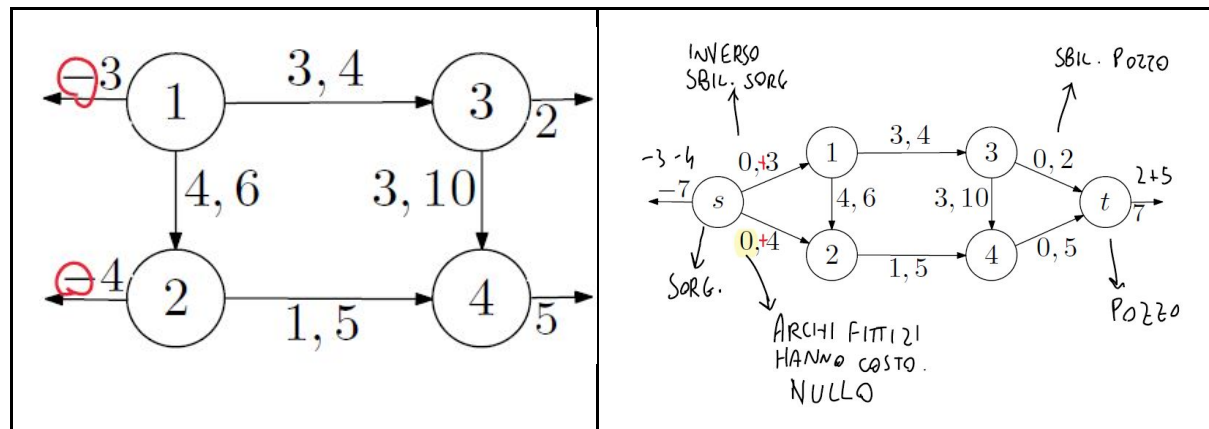
PER CATTURARE
I VINCOLI

dove

- ▶ $c \in \mathbb{R}^{|A|}$ è il **vettore dei costi**;
- ▶ $u \in \mathbb{R}^{|A|}$ è il **vettore delle capacità**;
- ▶ $E \in \mathbb{R}^{|N| \times |A|}$ è una sorta di **matrice di incidenza tra nodi e archi**;
- ▶ $b \in \mathbb{R}^{|N|}$ è il **vettore degli sbilanciamenti**.

Rilassamenti:

- Assumere che ci sia una sola sorgente e un solo pozzo
- Un problema generico può essere trasformato in un problema con una sola sorgente e un solo pozzo :
- Aggiungendo due nodi fittizi corrispondenti alla sorgente e al pozzo
 - Aggiungendo degli archi fittizi dalla sorgente alle sorgenti di partenza con costo nullo e capacità superiore pari all'inverso dello sbilanciamento della sorgente
 - Aggiungendo archi fittizi da ciascuno dei pozzi della rete di partenza al pozzo con costo nullo, ma capacità superiore pari allo sbilanciamento del pozzo
 - Lo sbilanciamento dell'unica sorgente sarà uguale alla somma degli sbilanciamenti delle sorgenti della rete di partenza. Similmente per i pozzi.



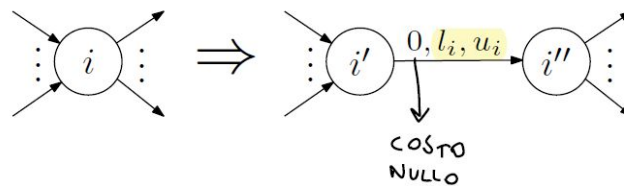
Data una rete G , si può costruire una rete H che sia in un certo senso equivalente a G ma che abbia capacità inferiori nulle. Per ogni arco $(i, j) \in A$:

- Si sottrae la quantità l_{ij} a b_j e a u_{ij} ;
- Si aggiunge la quantità l_{ij} a b_i .

- Occorrerà aggiungere alla funzione obiettivo la quantità $\sum_{(i,j) \in A} c_{ij} * l_{ij}$
 - In altre parole, ad un flusso x_{ij} in H corrisponderà un flusso $x_{ij} + l_{ij}$ in G.

Alcune volte è utile imporre che anche i nodi abbiano delle capacità, ossia che solo una quantità di flusso compresa nell'intervallo chiuso $[l_i; u_i]$ possa passare per il nodo $i \in N$. modelliamo questa situazione sdoppiando ciascun nodo i in due nodi i' ; i'' , in modo che:

- Tutti gli archi entranti in i vadano a finire in i' .
- Tutti gli archi uscenti da i partano da i'' .
- Vi sia un arco fittizio (i', i'') con costo nullo, capacità inferiore l_i e capacità superiore u_i .



PROBLEMA DEL FLUSSO MASSIMO (MF)

Possiamo vedere questo problema come una restrizione di MCF, dove cambiamo la funzione obiettivo: invece di minimizzare i costi si vuole massimizzare i flussi.

Data una rete $G = (N, A)$ fissiamo due nodi, una sorgente s e una destinazione t e vogliamo massimizzare il flusso da s a t ossia trovare il massimo valore v tale che:

- Se $b_s = -v * b_t = v$
- Se $b_i = 0$ in tutti gli altri casi

Allora esiste un flusso ammissibile di qualsiasi costo, un valore v ammissibile si dice valore del flusso x . Il problema è formalizzabile in PLI.

VINCOLI

$$\begin{array}{l}
 \text{ESCE} \qquad \qquad \text{ENTRA} \\
 \max v \\
 \text{IN } s \quad \sum_{(j,s) \in BS(s)} x_{js} + v = \sum_{(s,j) \in FS(s)} x_{sj}; \quad \text{IN} > \text{OUT} \\
 \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{Ciò che entra è uguale a ciò che esce} + v \\
 \sum_{(j,i) \in BS(i)} x_{ji} - \sum_{(i,j) \in FS(i)} x_{ij} = 0, \quad i \in N - \{s, t\}; \\
 \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \downarrow \text{IN} = \text{OUT} \\
 \text{IN } t \quad \sum_{(j,t) \in BS(t)} x_{jt} = \sum_{(t,j) \in FS(t)} x_{tj} + v; \quad \text{IN} < \text{OUT} \\
 \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{Ciò che esce è uguale a ciò che entra} + v \\
 0 \leq x_{ij} \leq u_{ij}, \quad (i, j) \in A.
 \end{array}$$

Come anticipato, il problema MF può essere visto come un caso particolare di MCF dove:

- I costi sono nulli
- Gli sbilanciamenti sono nulli
- Si aggiunge un arco fittizio da t ad s con costo -1 e capacità infinita
- Tutti i nodi diventano di trasferimento quindi minimizzare il costo equivale a massimizzare il flusso

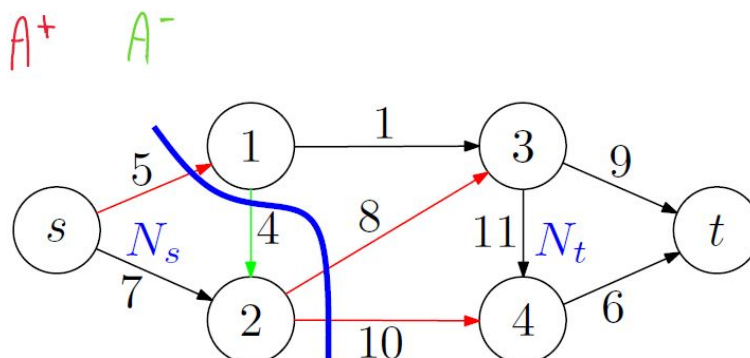
TAGLI

Un taglio in una rete $G = (N, A)$ è dato da una coppia (N', N'') di sottoinsiemi di N tali che $N' \cap N'' = \emptyset$ (no elementi in comune) e $N' \cup N'' = N$ (unendo i tagli si ottiene tutto N).

Un **(s, t)-taglio** è un taglio (N_s, N_t) dove $s \in N_s$ e $t \in N_t$.

Dato un (s, t) -taglio in $G = (N, A)$ abbiamo due sottoinsiemi di A (frontiera del taglio):

- $A^+(N_s, N_t) = \{(i, j) \in A \mid i \in N_s \text{ and } j \in N_t\} \rightarrow$ nella direzione del flusso
- $A^-(N_s, N_t) = \{(i, j) \in A \mid i \in N_t \text{ and } j \in N_s\} \rightarrow$ nella direzione opposta al flusso



Lemma: per ogni (s, t) -taglio (N_s, N_t) e ogni flusso ammissibile x con valore v :

$$1. v = \sum_{(i,j) \in A^+(N_s, N_t)} x_{ij} - \sum_{(i,j) \in A^-(N_s, N_t)} x_{ij}$$

$$2. v \leq \sum_{(i,j) \in A^+(N_s, N_t)} u_{ij}.$$

1) Il flusso v da s a t è uguale al (flusso in A^+ meno il flusso in A^-), ovvero il flusso lungo il taglio.

Dimostrazione:

$$v = \sum_{(s,j) \in A} x_{sj} - \sum_{(i,s) \in A} x_{is} = \sum_{k \in N_s} \left(\sum_{(k,j) \in A} x_{kj} - \sum_{(i,k) \in A} x_{ik} \right)$$

\downarrow DI TRASFERIM.

$$= \sum_{(i,j) \in A^+(N_s, N_t)} x_{ij} - \sum_{(i,j) \in A^-(N_s, N_t)} x_{ij}$$

- Il flusso v da s a t è uguale al flusso sugli archi da s a j meno il flusso da i a s .
 - Consideriamo ora dei generici nodi k in N_s : abbiamo che il flusso sugli archi di A da k a j meno il flusso da i a k resta uguale a v (i due flussi sono nulli per tutti i nodi diversi da s).
 - Infine analizziamo la frontiera del taglio e otteniamo che il flusso da i a j in A^+ meno il flusso da i a j in A^- (detto **flusso del taglio**) resta equivalente a v (i nodi diversi da s restano di trasferimento)
- 2) v è minore o uguale della capacità degli archi in A^+ (**capacità del taglio** $u(N_s, N_t)$)
 Dimostrazione: intuitivamente, $x_{ij} \leq u_{ij}$

Riassumendo questo lemma abbiamo che: $v = x(N_s; N_t) \leq u(N_s; N_t)$

Ovvero: il valore di un flusso ammissibile è sempre minore o uguale della capacità di qualunque taglio

GRAFI RESIDUI

Data una rete $G = (N_G; A_G)$ e un flusso ammissibile x , il grafo residuo G_x è il multigrafo $(N_{G_x}; A_{G_x})$ tale che:

- $N_{G_x} = N_G$ (i nodi del grafo residuo sono quelli del grafo di partenza);
- Gli archi in A_{G_x} sono di due tipi:
 - Per ogni arco $(i, j) \in A$ tale che $x_{ij} < u_{ij}$ esiste un arco da i a j in G_x (detto arco concorde);
 - Per ogni arco $(i, j) \in A$ tale che $x_{ij} > 0$ esiste un arco da j a i in G_x (detto arco discorde).
- Osserviamo come in N_{G_x} ci possano essere due archi da uno stesso nodo i ad uno stesso nodo j

CAMMINI AUMENTANTI

Un cammino aumentante P in una rete G rispetto a x non è nient'altro che un cammino semplice e orientato da s a t in G_x .

Definiamo P^+ l'insieme degli archi concordi in P , e P^- l'insieme dei suoi archi discordi.

Dato un cammino aumentante P rispetto a x , definiamo la capacità di P rispetto a x come

$$\theta(P; x) = \min\{\min\{u_{ij} - x_{ij} \mid (i, j) \in P^+\}, \min\{x_{ij} \mid (j, i) \in P^-\}\}$$

Dato un flusso x , un cammino P in G_x e un reale θ , definiamo $x(P; \theta)$ il flusso definito come segue:

$$(x(P, \theta))_{ij} = \begin{cases} x_{ij} + \theta & \text{se } (i, j) \in P^+; \\ x_{ij} - \theta & \text{se } (j, i) \in P^-; \\ x_{ij} & \text{altrimenti.} \end{cases}$$

FORD FULKERSON (teoremi e complessità)

- 1) $x = 0$ (flusso nullo quindi ammissibile);
- 2) Costruisci G_x e determina se G_x ha un cammino aumentante P .
 - a) In caso P non esista, termina e restituisci x ;
- 3) $x = x(P; \theta(P; x)) \rightarrow$ update di x
- 4) Ritorna al punto 2.

Lemma: Se x è ammissibile, allora anche $x(P, \theta(P; x))$ è ammissibile.

Lemma: Se x è un flusso ammissibile massimo, allora G_x non ha cammini aumentanti.

Dimostrazione: Se ci fosse un cammino aumentante in G_x , allora x non sarebbe massimo, perché sarebbe possibile aumentare il valore del flusso.

Lemma: Se G_x non ha cammini aumentanti, allora esiste un taglio di capacità pari a v .

Dimostrazione: Basta considerare il taglio $(N_s; N_t)$, dove N_s contiene tutti e soli i nodi raggiungibili da s in G_x (e $N_t = N - N_s$).

$$\begin{aligned}
 v = x(N_s, N_t) &= \left(\sum_{(i,j) \in A^+(N_s, N_t)} x_{ij} - \sum_{(i,j) \in A^-(N_s, N_t)} x_{ij} \right) \\
 &= \sum_{(i,j) \in A^+(N_s, N_t)} u_{ij} - \sum_{(i,j) \in A^-(N_s, N_t)} 0 \\
 &= \underline{u(N_s, N_t)}
 \end{aligned}$$

FLUSSO
 LUNGO
 TAGLIO

CAPACITÀ

Infatti il flusso lungo il taglio è uguale al flusso in N_s e N_t che a sua volta equivale ai flussi sugli archi in A^+ meno il flusso in A^- .

Passiamo alle capacità: dato che il flusso è massimo sappiamo che la capacità in A^+ meno quella in A^- resta uguale alla capacità in A^+ perchè non c'è flusso da t ad s .

Quindi il flusso da s a t è uguale alla capacità dei nodi nel taglio N_s, N_t :

Teorema (correttezza parziale)

Se l'algoritmo di Ford-Fulkerson termina, allora il flusso x è un flusso massimo.

Dimostrazione

- 1) Se Ford-Fulkerson termina, allora G_x non ha cammini aumentanti.
 - a) Ma quindi per il lemma precedente esiste un taglio di capacità pari a v .
 - b) E a questo punto v non può essere che massimo, perché se non lo fosse avremmo un taglio di capacità inferiore al valore di un flusso ammissibile, ovvero potremmo aumentare il flusso.

Teorema: Il valore del massimo flusso è uguale alla minima capacità dei tagli.

Dimostrazione: Basta dimostrare che il valore del massimo flusso è maggiore o uguale alla capacità di un taglio.

Ma se x è ammissibile e massimo, G_x non ha cammini aumentanti, e quindi esiste un taglio di capacità pari a v .

Teorema: Se le capacità di G sono numeri interi, allora esiste almeno un flusso intero massimo.

Dimostrazione:

- 1) Se le capacità sono intere, allora il flusso massimo sarà al più nU dove $U = \max\{u_{ij} \mid (i; j) \in A\}$.
- 2) Si parte da un flusso intero, e l'interezza è preservata, perché per ogni cammino aumentante P , $(P; x)$ è un numero intero.
 - a) Di conseguenza, il valore del flusso aumenta almeno di 1 ad ogni iterata.
- 3) L'algoritmo terminerà quindi dopo al più nU iterate.

Quindi se le capacità sono intere, allora la complessità è $O(mnU)$, ovvero è pseudopolinomiale.

EDMONDS-KARP

L'Algoritmo di Edmonds-Karp non è nient'altro che l'algoritmo di Ford-Fulkerson dove, però, la ricerca del cammino aumentante viene eseguita visitando in ampiezza (BFS) il grafo residuo G_x , in questo modo i cammini aumentanti saranno sempre cammini di lunghezza minima.

L'algoritmo è corretto in quanto caso particolare di Ford-Fulkerson, per calcolarne la complessità si procede invece come segue:

Se in FF i cammini aumentanti sono di lunghezza minima, allora la distanza di un generico nodo i dalla sorgente s in G_x non può diminuire.

Da ciò si deduce che il numero di iterazioni di EK non può, asintoticamente essere più grande di $N \cdot A$.

Con $\delta_x(i, j)$ indichiamo la distanza tra i nodi i, j in G_x

Lemma: Se, durante l'esecuzione di EK, il flusso y è ottenuto da x tramite un'operazione di aumento del flusso in un cammino aumentante, allora per ogni nodo $i \in N$, vale che $\delta_x(s, i) \leq \delta_y(s, i)$.

Ovvero la distanza tra s e i nel grafo residuo x è minore o uguale alla distanza tra s e i nel grafo residuo y .

Teorema

Il numero di iterazioni di EK è $O(NA)$, quindi la sua complessità è $O(NA^2)$.

Dimostrazione

Un arco (i, j) in G_x è detto critico per un cammino aumentante P se la sua capacità è uguale a $\theta(P, x)$. In ogni cammino aumentante, esiste almeno un arco critico. Una volta saturato l'arco critico questo sparisce dal grafo residuo.

Si può dimostrare che dati i, j connessi da un arco in A , il numero di volte in cui (i, j) è critico è al più $O(N)$, e visto che di tali coppie ne esistono al più $O(A)$, in totale potremo avere al più $O(NA)$ iterazioni.

Quando (i, j) diventa critico la prima volta, deve valere che $\delta_x(s, j) = \delta_x(s, i) + 1$ (ovvero i e j si trovano ad un solo passo di distanza), dove x è il flusso, e a quel punto sparisce dal grafo residuo.

L'unico modo per ricomparirvi è fare in modo che il flusso (reale o virtuale) da i a j diminuisca, e questo vuol dire che $\delta_y(s, i) = \delta_y(s, j) + 1$, dove y è il flusso.

$$\text{Dunque: } \delta_y(s; i) = \delta_y(s; j) + 1 \geq \delta_x(s; j) + 1 = \delta_x(s; i) + 2$$

(la distanza tra s ed i nel grafo y è uguale alla distanza tra s e $j + 1$ ed è maggiore o uguale alla distanza tra s e j nel grafo x , che a sua volta è uguale alla distanza tra s e i più due) Di conseguenza, da un momento in cui (i, j) diventa critico al successivo, la sua distanza da s aumenta di almeno 2. Siccome tale distanza non può essere superiore a $|N|$, il numero di volte in cui (i, j) può diventare critico è lineare in $|N|$.

GOLDBERG - TARJAN

Riesce ad abbattere la complessità di EK portandola a $O(NA^2)$.

Un **preflusso** è un vettore x tale che bilancia i vincoli di capacità ma non quelli di bilanciamento:

$$\ell_i \rightarrow \left\{ \begin{array}{l} \sum_{(j,i) \in BS(i)} x_{ji} - \sum_{(i,j) \in FS(i)} x_{ij} \geq 0, \quad i \in N - \{s, t\}; \\ 0 \leq x_{ij} \leq u_{ij}, \quad (i, j) \in A. \end{array} \right.$$

Se l'eccesso di un nodo è nullo allora è bilanciato, altrimenti è attivo

Etichettatura

Vettore che indica la distanza del nodo i da t (può avere solo valori R^+)

Data una etichettatura, un arco (i, j) è detto AMMISSIBILE per i sse NON SATURO, mentre per j è detto AMMISSIBILE sse NON VUOTO

Push

Se i è un nodo attivo e esiste un arco (i, j) ammissibile per esso, allora è possibile inviare l'eccesso, o una parte di esso, lungo l'arco (PUSH FORWARD)

Se i è un nodo attivo e esiste un arco (j, i) ammissibile per esso, allora è possibile inviare l'eccesso, o una parte di esso, lungo l'arco (PUSH BACKWARD)

Relabel

Supponiamo che il nodo i non abbia nodi incidenti che siano ammissibili. In tal caso l'unica strada percorribile è quella di aumentare l'etichetta di i . Così facendo rendiamo ammissibile un arco incidente ad i

GOLDBERG TARJAN(G, s, t)

1. $x \leftarrow 0$;
2. $x_{sj} \leftarrow u_{sj} \quad \forall (s, j) \in FS(s)$;
3. $d \leftarrow ETICHETTATURAVALIDA(G)$;
4. $d_s \leftarrow n$;
5. Se tutti i nodi (diversi da s e t) sono bilanciati, allora termina e restituisci x .
6. Sia v un qualunque nodo sbilanciato \rightarrow SE NON CI SONO VA A 5
7. Se esiste (v, j) ammissibile per v , allora esegui PUSHFORWARD(v, j) e torna a 6, altrimenti prosegui
8. se esiste (i, v) ammissibile per v , allora esegui PUSHBACKWARD(i, v) e torna a 6, altrimenti prosegui
9. esegui RELABEL(v) e torna a 6.

PROBLEMA DEL FLUSSO MINIMO

Pseudoflusso

Uno pseudoflusso è un vettore x che soddisfa i vincoli di capacità, ossia tale che $0 \leq x_{ij} \leq u_{ij}$ con $i, j \in A$

Dato uno pseudoflusso x , i nodi sbilanciati rispetto a x fanno parte di uno dei seguenti due insiemi:

$$\left. \begin{aligned} O_x &= \{i \in N \mid e_x(i) > 0\}; \\ D_x &= \{i \in N \mid e_x(i) < 0\}. \end{aligned} \right\} \text{SBILANCIAMENTO}$$

I nodi in O_x sono detti nodi con eccesso di flusso, mentre quelli in D_x sono detti nodi con difetto di flusso.

► Lo **sbilanciamento complessivo** di x è definito come:

$$g(x) = \sum_{i \in O_x} e_x(i) = - \sum_{j \in D_x} e_x(j)$$

Pseudoflusso

Quando si lavora con pseudoflussi, la nozione stessa di cammino aumentante diventa più generale e ogni arco avrà ora un costo: in un arco concorde (i, j) di G_x , il costo è semplicemente c_{ij} , mentre in un arco discorde (j, i) di G_x , il costo è invece $-c_{ij}$.

Un cammino P tra i e j in G_x verrà quindi detto cammino aumentante tra i e j .

I suoi archi possono essere partizionati in P^+ e P^- .

La sua capacità $\theta(P, x)$ è definita come al solito.

Un cammino aumentante tra i e i viene anche detto **ciclo aumentante**.

Dato uno pseudoflusso x e un cammino aumentante P , è possibile inviare $0 \leq \theta \leq \theta(P, x)$ unità di flusso lungo P , attraverso l'operazione $x(P, \theta)$, che conosciamo già e lo indicheremo con

$x \oplus P \theta$.

Se P è un cammino aumentante da i a j in G_x , allora lo pseudoflusso $x(P, \theta)$ avrà gli stessi sbilanciamenti di x (tranne in i e j) \rightarrow se $i = j$ il vettore degli sbilanciamenti resta inalterato.

Il costo di un cammino aumentante è definito come $c(P) = \sum_{(i,j) \in P^+} c_{ij} - \sum_{(i,j) \in P^-} c_{ij}$

Teorema (Struttura degli Pseudoflussi)

Siano x e y due pseudoflussi qualunque. Allora esistono $k \leq n + m$ cammini aumentanti P_1, \dots, P_k , tutti per x , di cui al più m sono cicli, tali che

$$\begin{aligned}z_1 &= x \\z_{i+1} &= z_i \oplus \theta_i P_i \quad 1 \leq i \leq k \\z_{k+1} &= y \\0 &\leq \theta_i \leq \theta(P_i, z_i).\end{aligned}$$

Inoltre, tutti i P_i hanno come estremi dei nodi in cui lo sbilanciamento di x è diverso da quello di y .

Pseudoflussi minimiali

A differenza di quello che succede in MF, in MCF non possiamo permetterci di aumentare il flusso indiscriminatamente, quindi bisogna determinare quali siano le operazioni di aumento lecite e quali siano le proprietà sui flussi che esse garantiscano, ed è qui che introduciamo la nozione di **pseudoflusso minimale**: un pseudoflusso x che abbia costo minimo tra tutti gli pseudoflussi aventi lo stesso vettore di sbilanciamento e_x .

Lemma:

Uno pseudoflusso ammissibile è minimale sse non esistono cicli aumentanti di costo negativo

Dimostrazione:

- Per contrapposizione: se esiste un ciclo aumentante di costo negativo in G_x , applicarlo fa diminuire il costo senza alterare lo sbilanciamento, in contraddizione con la minimalità di x .
- ← Ancora per contrapposizione: supponiamo che x non sia minimale, ossia che esista y con $c y < c x$ e $e_y = e_x$. Allora per il teorema sugli pseudoflussi possiamo scrivere $y = x \oplus \theta_1 P_1 \oplus \dots \oplus \theta_n P_n$, dove $\theta_i > 0$ e ciascun P_i è un ciclo. Da $c y < c x$ discende però che $c x > c x + \theta_1 c(P_1) + \dots + \theta_n c(P_n)$ e quindi che $c(P_i) < 0$ per qualche i .

Teorema:

Sia x un pseudoflusso minimale e sia P un cammino aumentante rispetto ad x avente costo minimo tra tutti i cammini che uniscono un nodo di O_x ad un nodo di D_x . Allora, qualunque sia $\theta \leq \theta(x, P)$, abbiamo che $x(\theta, P) = x \oplus \theta P$ è ancora pseudoflusso minimale.

Dimostrazione:

Siano s e t i vertici che P collega. Supponiamo che $\theta \leq \theta(x, P)$ e che y sia un qualunque pseudoflusso con vettore di sbilanciamento $e_x(\theta, P)$. Per il Teorema sulla struttura degli pseudoflussi esistono: k cammini aumentanti P_1, \dots, P_k rispetto a x , tutti da s a t ; h cicli aumentanti C_1, \dots, C_h rispetto a x , tali che $y = x \oplus \theta_1 P_1 \oplus \dots \oplus \theta_k P_k \oplus \mu_1 C_1 \oplus \dots \oplus \mu_h C_h$, (dove tutti gli θ_i, μ_j sono positivi).

Deve essere, per ragioni che hanno a che fare con lo sbilanciamento, che $\theta_i = \theta$. Poiché x è minimale, $c(C_i) \geq 0$.

Siccome P ha costo minimo, $c(P_i) \geq c(P)$.

//la somma dei cammini aumentanti dello pseudoflusso, deve essere inferiore al cammino aumentante che stiamo considerando, lo stesso vale della capacità residua. per i cicli ogni cammino aumentante deve avere costo > 0 , altrimenti è possibile diminuire quel cammino

ALGORITMI AUSILIARI

È abbastanza facile costruire uno pseudoflusso minimale x , se non si bada agli sbilanciamenti (ad esempio x_{ij} potrebbe valere 0 se $c_{ij} \geq 0$ oppure u_{ij} altrimenti, in modo da non avere cicli negativi e x minimale)

CAMMINI MINIMI SUCCESSIVI

1. $x \leftarrow \text{PSEUDOFUSSO MINIMALE}(G)$;
2. Se $g(x) = 0$, allora termina e restituisci x ;
3. Cerca un cammino di costo minimo P tra un nodo di $i \in O_x$ e $j \in D_x$; se non esiste, termina: il problema è vuoto;
4. $x \leftarrow x(P, \min\{\theta(P, x), e_x(i), -e_x(j)\})$;
5. Torna al punto 2.

Ad ogni passo, il flusso x rimane minimale. Se l'algoritmo termina, allora lo sbilanciamento $g(x) = 0$ e quindi x è uno pseudoflusso minimale con sbilanciamento nullo, ossia un flusso di costo minimo.

Riguardo la terminazione, se b e u sono vettori di numeri interi, possiamo osservare che:

- Lo pseudoflusso iniziale è esso stesso intero;
- Se x è pseudoflusso intero, allora la capacità $(x; P)$ rimane intera;
- Lo pseudoflusso x rimane quindi sempre intero.
- Ad ogni passo, $g(x)$ diminuisce di almeno 1

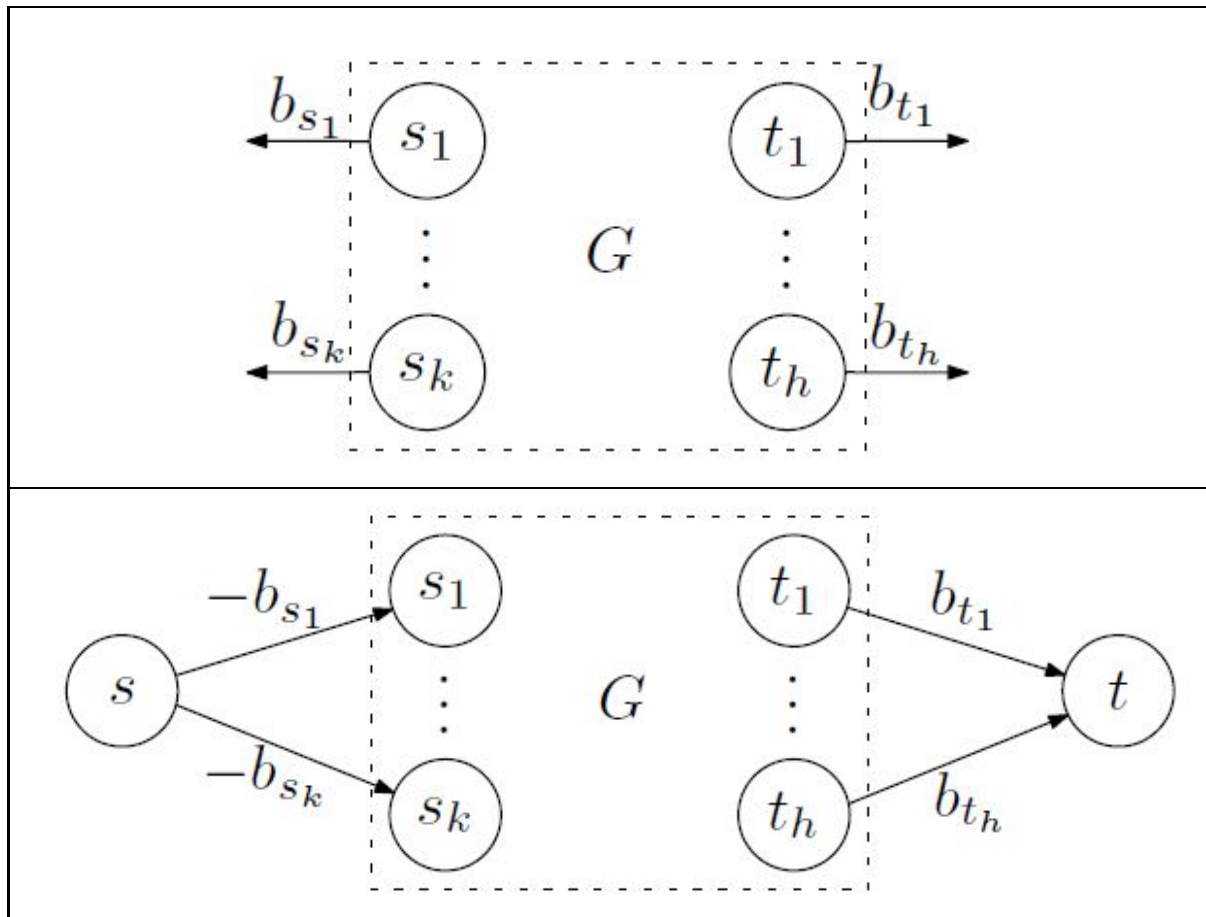
Lo sbilanciamento iniziale g^* è al più: $g^* \leq \sum_{b_i > 0} b_i + \sum_{c_{ij} < 0} u_{ij}$ e diminuisce di almeno 1 ad ogni iterazione \rightarrow avremo al massimo g^* iterazioni il cui costo computazionale è dominato

dalla ricerca di un cammino minimo, che possiamo eseguire in tempo $O(NA)$.

Nel caso peggiore la complessità sarà quindi $O(g^*NA)$, ovvero pseudopolinomiale nella dimensione del grafo.

COSTRUIRE UN FLUSSO AMMISSIBILE

Data una rete G , è possibile determinare se esiste un flusso ammissibile risolvendo il problema MF sulla rete ottenuta nel modo seguente:



Otteniamo così un algoritmo, che chiamiamo FlussoAmmissibile(G), che data una rete, calcola un flusso ammissibile per essa (se esiste).

CANCELLAZIONE DEI CICLI

1. Se FlussoAmmissibile(G) restituisce un flusso ammissibile, allora mettilo in x , altrimenti termina: il problema è vuoto.
2. Cerca un ciclo di costo negativo in G_x . Se non lo trovi, allora termina e restituisci x , altrimenti metti il ciclo in C .
3. $x \leftarrow x(C, \theta(C, x))$;
4. torna al punto 2.

La correttezza dell'algoritmo è una banale conseguenza del Lemma (che abbiamo dimostrato) sull'equivalenza tra assenza di cicli aumentanti e ottimalità.

Come al solito, se le capacità sono numeri interi, allora qualcosa diminuisce di almeno 1 ad ogni iterazione, ossia il costo (e quindi l'algoritmo termina).

Il costo di qualunque flusso ammissibile è compreso tra $-Au^*c^*$ e Au^*c^* , dove

$$\bar{u} = \max\{u_{ij} \mid (i, j) \in N\};$$
$$\bar{c} = \max\{c_{ij} \mid (i, j) \in N\}.$$

La complessità dell'algoritmo sarà quindi pseudopolinomiale, ossia:

$$O(NA) \cdot O(Au^*c^*) = O(NA^2u^*c^*)$$

PROBLEMI DI ACCOPPIAMENTO

Nei problemi di accoppiamento, si lavora con grafi bipartiti non orientati ossia con grafi nella forma $G = (O \cup D, A)$, dove:

- O insieme nodi origine
- D insieme nodi destinazione
- A insieme archi che collegano i nodi O, D con associato un costo
- I nodi collegati con M sono detti **accoppiati**, gli altri vengono detti **esposti**

M è detto accoppiamento perfetto sse non vi sono nodi esposti

► Il *costo* di un accoppiamento M è nient'altro che

$$c(M) = \sum_{(i,j) \in M} c_{ij}.$$

L'arco di accoppiamento di costo massimo viene detto **arco di bottleneck** e il suo costo **valore di bottleneck**

Accoppiamento di Massima Cardinalità: si vuole determinare il massimo numero di archi che collegano nodi diversi (lo stesso nodo non può avere due archi selezionati collegati)

Accoppiamento di Costo Minimo: si vuole determinare l'accoppiamento di costo minimo tra tutti gli accoppiamenti perfetti

Accoppiamento di Massima Cardinalità Bottleneck: Tra tutti gli accoppiamenti di massima cardinalità, si vuole determinare quello con valore di bottleneck minimo

ACCOPPIAMENTO DI MASSIMA CARDINALITA'

Il problema può essere visto come un problema di flusso massimo con più sorgenti (O) e più pozzi (D):

- Le capacità saranno tutte pari a 1

- Ci interessano solamente flussi interi
- M sottoinsieme degli archi di A che non hanno nodi in comune e vengono detti **interni**, gli altri vengono detti **esterni**

Flussi ammissibili interi e accoppiamenti sono in corrispondenza biunivoca.

Indicheremo quindi, ad esempio, un flusso con M e il relativo grafo residuo con G_M .

Ogni **cammino aumentante** in G_M deve:

- Essere alternante, ossia consistere di archi interni, seguiti da archi esterni, seguiti da archi interni, e così via
- Partire da un'origine esposta e arrivare ad una destinazione esposta.
- In altre parole, se $PE = P - M$ e $PI = M \cap P$ sono rispettivamente gli archi esterni e interni di un cammino aumentante P , allora $|PE| - |PI| = 1$

La capacità $\theta(M, P)$ di un cammino aumentante P sarà sempre 1, e di conseguenza $M \oplus (\theta(M, P))P = (M - PI) \cup PE$

Otteniamo in questo modo un algoritmo del tutto simile a FF, ma in cui la ricerca del cammino aumentante può essere eseguita tramite una semplice procedura di visita, con complessità $O(mn)$

Nel caso il problema in questione sia l'**accoppiamento di costo minimo**, si può procedere similmente all'accoppiamento di massima cardinalità, ma in questo caso specializzando gli algoritmi per MCF:

In particolare, i cammini minimi aumentanti potrebbero essere visti come cammini esposti tra due vertici esposti, rispettivamente in O e in D .

MODULO 3

Nozioni preliminari

- Iperpiano
 - Insieme $\{x \in \mathbb{R}^n \mid ax = b\}$ delle soluzioni dell'equazione lineare $ax = b$, dove $a \in \mathbb{R}^n$ e $b \in \mathbb{R}$.
- Semispazio
 - Insieme $\{x \in \mathbb{R}^n \mid ax \leq b\}$ delle soluzioni dell'equazione lineare $ax \leq b$, dove $a \in \mathbb{R}^n$ e $b \in \mathbb{R}$;
 - Un iperpiano è il "confine" del corrispondente semispazio.
- Poliedro
 - Intersezione P di un numero finito m di semispazi.
 - Devono esistere una matrice $A \in \mathbb{R}^{m \times n}$ e un vettore $b \in \mathbb{R}^m$ tali che $P = \{x \mid Ax \leq b\}$.
- Insieme Convesso
 - Un insieme $C \subseteq \mathbb{R}^n$ tale che tutti i punti che connettono $x, y \in C$ sono anch'essi in C , ossia
$$\forall x, y \in C \quad \forall \alpha \in [0, 1] \quad \alpha x + (1 - \alpha)y \in C.$$
 - Semispazi e poliedri sono insiemi convessi.
- Se consideriamo il poliedro $P = \{x \mid Ax \leq b\}$ (dove $A \in \mathbb{R}^{m \times n}$) e fissiamo un qualunque sottoinsieme I di $\{1, \dots, m\}$, indichiamo:
 - Con I^* il complementare $\{1, \dots, m\} - I$ di I .
 - Con A_I la sottomatrice di A ottenuta considerando solo le righe con indice in I
 - Con P_I il poliedro definito come segue:
$$\{x \mid A_I x = b_I \wedge A_{I^*} x \leq b_{I^*}\}$$
- Faccia
 - I Se I è tale che P_I non è vuoto, chiamiamo il poliedro P_I faccia di P .
 - Il numero di facce distinte di un poliedro $\{x \mid Ax \leq b\}$, dove $A \in \mathbb{R}^{m \times n}$, è al più pari a 2^m .
 - P stesso è una faccia, ovvero P_\emptyset
 - Le facce proprie (cioè non banali) e massimali sono dette faccette.
 - La dimensione di una faccia è la dimensione del più piccolo sottospazio che la contenga.

Vertici

- Una faccia determinata da una matrice A_I di rango k ha dimensione $n - k$ o inferiore.
 - Può essere inferiore a causa delle equazioni in A_I , che possono contenere un'equazione implicita.
- Le facce determinate da matrici A_I di rango n hanno quindi, necessariamente, dimensione 0 e sono dette vertici.

- Le facce determinate da matrici A_i di rango n hanno quindi, necessariamente, dimensione 0 e sono dette vertici.
- D'altra parte, le facce sono sempre non-vuote.
- Le facce individuate da sottomatrici A_i di rango $n - 1$ hanno dimensione al più 1 e sono dette spigoli.

Soluzioni di Base

- Supponiamo che B sia tale che A_B sia matrice quadrata e invertibile. Allora:
 - B è detta base;
 - AB è detta matrice di base;
 - Una soluzione di base x_B tale che $x_B \in P$ è detta ammissibile, altrimenti non ammissibile
 - È facile rendersi conto che i vertici di P sono tutte e sole le sue soluzioni di base ammissibili.

Vincoli Attivi

- Se $x \in P$, allora i vincoli che vengono soddisfatti come uguaglianze, sono detti attivi in x .
- Indichiamo con $I(x)$ l'insieme degli indici dei vincoli attivi in x :

$$I(x) = \{i \mid A_i x = b_i\}$$

- Per ogni $J \subseteq I(x)$, l'insieme P_J è una faccia di P , e $P_{I(x)}$ è la faccia minimale tra esse.
-

Inviluppi Convessi

- I poliedri possono essere rappresentati per facce, come abbiamo fatto fin'ora, ma anche per punti ossia facendo leva sull'insieme dei vertici
- Dato un insieme di punti $X = \{x_1, \dots, x_s\} \subseteq \mathbb{R}^n$, l'inviluppo convesso di X è definito come l'insieme

$$\text{conv}(X) = \left\{ x = \sum_{i=1}^s \lambda_i x_i \mid \sum_{i=1}^s \lambda_i = 1 \wedge \lambda_i \geq 0 \right\}$$

- Si può dimostrare che $\text{conv}(X)$ è il più piccolo insieme convesso che contiene tutti i punti di X .
- $\text{conv}(X)$ è un politopo, ossia un poliedro limitato, i cui vertici sono tutti in X .
 - Non tutti i poliedri sono politopi, perché i poliedri possono essere illimitati.

Coni Convessi

- Un insieme $C \subseteq \mathbb{R}^n$ è detto cono sse per ogni $x \in C$ e per ogni $\alpha \in \mathbb{R}^+$ vale che $\alpha x \in C$.
- I coni che siano anche insiemi convessi (detti coni convessi) sono caratterizzabili equivalentemente come gli insiemi C tali che

$$x, y \in C \wedge \lambda, \mu \in \mathbb{R} \quad \Rightarrow \quad \lambda x + \mu y \in C$$

- Anche per i coni convessi esiste una rappresentazione basata sulle direzioni: dato un insieme $V = \{v_1, \dots, v_t\} \subset \mathbb{R}^n$, il cono finitamente generato da V è

$$\text{cono}(V) = \left\{ v = \sum_{i=1}^t \nu_i v_i \mid \nu_i \in \mathbb{R}^+ \right\}$$

- Si può dimostrare che $\text{cono}(V)$ è il più piccolo cono convesso che contiene tutti i vettori di V .

Teorema di Motzkin

- Dati $X, Y \subseteq \mathbb{R}^n$, indichiamo con $X + Y$ il sottoinsieme di \mathbb{R}^n definito ponendo

$$X + Y = \{x + y \mid x \in X \wedge y \in Y\}$$

Teorema (Motzkin) $P \subseteq \mathbb{R}^n$ è un poliedro sse esistono X, V finiti tali che

$$P = \text{conv}(X) + \text{cono}(V)$$

- Nel contesto del Teorema di Motzkin, diremo che P è generato dai punti in X e dalla direzioni in V .
- Se P è poliedro generato dai punti di X e X è minimale, allora i suoi elementi sono tutti e soli i vertici di P .
- Analogamente: se P è poliedro generato dalle direzioni in V e V è minimale, allora i suoi elementi sono detti raggi esterni e corrispondono alle direzioni degli spigoli illimitati.
- Due rappresentazioni:
 1. Poliedri come **intersezioni di semispazi**.
 2. Poliedri come **somma di un politopo e di un cono**.
- Le due rappresentazioni sono equivalenti (grazie al teorema di Motzkin) ma non hanno la stessa dimensione.
 - Prendiamo come controesempio il poliedro definito dall'insieme di vincoli

$$0 \leq x_1 \leq 1 \quad 0 \leq x_2 \leq 1 \quad \dots \quad 0 \leq x_n \leq 1$$
 - I semispazi coinvolti sono $2n$.
 - vertici sono invece 2^n ; per rendersene conto basta osservare che i poliedri definiti sono gli ipercubi in \mathbb{R}^n di lato pari a 1 e con un vertice nell'origine. Tali ipercubi hanno effettivamente 2^n vertici.

Teorema

Sia $P = \{x \mid Ax \leq b\}$ e siano $x_1, \dots, x_s, v_1, \dots, v_t \in \mathbb{R}^n$ tali che

$$P = \text{conv}(\{x_1, \dots, x_s\}) + \text{cono}(\{v_1, \dots, v_t\})$$

Allora il problema $\max\{cx \mid Ax \leq b\}$ ha ottimo finito sse $cv_j \leq 0$ per ogni $j \in \{1, \dots, t\}$. In tal caso esiste inoltre un $k \in \{1, \dots, s\}$ tale che x_k è una soluzione ottima.

Tale problema ha ottimo finito sse $cv_j \leq 0$ per ogni $j \in \{1, \dots, t\}$. Infatti:

\Rightarrow Se fosse $cv_j > 0$ per qualche $j \in \{1, \dots, t\}$, allora si potrebbe pompare v_j facendo crescere a piacimento la funzione obiettivo.

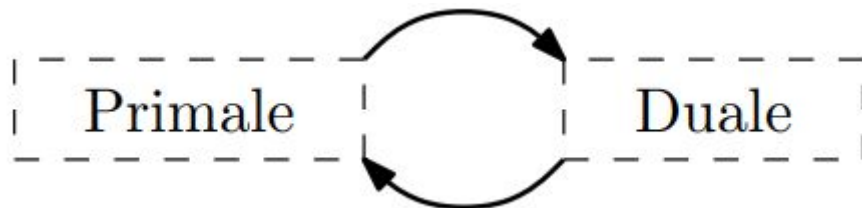
\Leftarrow Supponiamo che $cv_j \leq 0$ per ogni $j \in \{1, \dots, t\}$, e prendiamo un $y \in P$. Abbiamo che, se λ_i e ν_j sono i corrispondenti coefficienti del teorema di decomposizione,

$$\begin{aligned}
 cy &= \sum_{i=1}^s \lambda_i(cx_i) + \sum_{j=1}^t \nu_j(cv_j) \\
 &\leq \sum_{i=1}^s \lambda_i(cx_i) \leq \sum_{i=1}^s \lambda_i(cx_k) = cx_k
 \end{aligned}$$

dove x_k è il vettore tale che $x_k = \max\{cx_i \mid i = 1, \dots, s\}$. Quindi x_k è una soluzione ottima finita.

Dualità

- La teoria della dualità è una branca dell'algebra lineare che risulta estremamente utile nella costruzione degli algoritmi per PL.
- In questa parte del corso, daremo solo uno sguardo alla teoria della dualità, senza addentrarci troppo nei dettagli.
- La teoria della dualità si basa sulla definizione di un'involuzione (ossia di una funzione inversa di sé stessa) che mappa ogni problema PL nel suo duale:



Esempio: Problema di Trasporto.

Primale e Duale

- Lavoreremo con **coppie asimmetriche**:
 - Primale: $\max\{cx \mid Ax \leq b\}$;
 - Duale: $\min\{yb \mid (yA = c) \wedge (y \geq 0)\}$.
- Esiste anche il concetto di **coppie simmetriche**:
 - Primale: $\max\{cx \mid (Ax \leq b) \wedge (x \geq 0)\}$;

- Duale: $\min\{yb \mid (yA \geq c) \wedge (y \geq 0)\}$.
- È abbastanza facile dimostrare che il duale del duale è il primale.
 - Ad esempio, nel caso di coppia simmetrica, possiamo esprimere il duale come

$$\begin{aligned}
 & - \max\{y(-b) \mid (yA \geq c) \wedge (y \geq 0)\} \\
 & = - \max\{(-b^T)y \mid ((-A^T)y \leq -c) \wedge (y \leq 0)\} \\
 & \text{il cui duale è} \\
 & - \min\{-cx \mid ((x(-A^T) \geq (-b)) \wedge (x \geq 0))\} \\
 & = \max\{cx \mid Ax \leq b \wedge (x \geq 0)\}
 \end{aligned}$$

Teorema Debole di Dualità

Se x e y sono soluzioni ammissibili per il primale e il duale, rispettivamente, allora $cx \leq yb$.

Dimostriamo il teorema nel caso della coppia asimmetrica:

$$\left. \begin{array}{l} Ax \leq b \\ \bar{y}A = c, \bar{y} \geq 0 \end{array} \right\} \implies \left. \begin{array}{l} \bar{y}Ax \leq \bar{y}b \\ \bar{y}Ax = c\bar{x} \end{array} \right\} \implies c\bar{x} \leq \bar{y}b$$

e nel caso della coppia simmetrica:

$$\left. \begin{array}{l} Ax \leq b, x \geq 0 \\ \bar{y}A \geq c, \bar{y} \geq 0 \end{array} \right\} \implies \left. \begin{array}{l} \bar{y}Ax \leq \bar{y}b \\ \bar{y}Ax \geq c\bar{x} \end{array} \right\} \implies c\bar{x} \leq \bar{y}b$$

Corollario

Se il primale è illimitato, allora il duale è vuoto.

Dimostrazione

Se il primale è illimitato, allora per ogni $M \in \mathbb{R}$ esiste una soluzione ammissibile x per il primale con $cx > M$. Ma quindi, se per assurdo ci fosse y ammissibile per il duale, troveremmo x ammissibile per il primale con $cx > yb$, in contrasto con il TdD.

Corollario

Se x^* e y^* sono soluzioni ammissibili per il primale e il duale, rispettivamente, e $cx^* = y^*b$, allora x^* e y^* sono soluzioni ottime.

Dimostrazione

Se $cx^* = y^*b$ e x non fosse ottima, troveremmo z ammissibile per il primale con $cz > cx^*$ e quindi con $cz > y^*b$, in contrasto con il TdD.

Direzioni Ammissibili

- Data una coppia asimmetrica, consideriamo una soluzione ammissibile x per il primale, e chiediamoci se spostandoci lungo una direzione dell'iperspazio a partire da x , si resta o meno nella regione ammissibile.
- Un vettore $\xi \in \mathbb{R}^n$ è detto Direzione Ammissibile se esiste $\lambda^* > 0$ tale che $x(\lambda) = x^* + \lambda \xi$ è ammissibile nel primale per ogni $\lambda \in [0, \lambda^*]$.

Lemma

Il vettore ξ è direzione ammissibile per x sse $A_i(x) \xi \leq 0$.

Dimostrazione.

- ▶ Un modo equivalente di definire ξ come direzione ammissibile è dire che per ogni $i \in \{1, \dots, m\}$,

$$A_i x(\lambda) = A_i \bar{x} + \lambda A_i \xi \leq b_i$$

- ▶ Osserviamo però che:
 - ▶ Se $i \in I(\bar{x})$, allora $A_i \bar{x} = b_i$, e quindi l'equazione è verificata se e solo se $\lambda A_i \xi \leq 0$.
 - ▶ se $i \notin I(\bar{x})$, allora l'equazione è verificata da qualunque ξ , purché λ sia piccolo a sufficienza.

Direzioni di Crescita

- Una direzione $\xi \in \mathbb{R}^n$ è una direzione di crescita per x^* se uno spostamento λ lungo ξ fa crescere il valore della funzione obiettivo, ossia se:
$$cx(\lambda) = cx^* + \lambda c \xi > cx^* \Leftrightarrow c \xi > 0$$
- La nozione di direzione di crescita, dunque, non dipende dal punto x^* !
- Osserviamo che:
 - Se $c = 0$, allora la funzione obiettivo vale sempre 0 e quindi tutte le soluzioni ammissibili sono ottime.
 - Se $c \neq 0$, allora se esiste una direzione ammissibile per x che sia anche di crescita, allora x non può essere ottimo.

L'ALGORITMO DEL SIMPLESSO

- Prima di presentare l'algoritmo, conviene dare uno sguardo alla sua struttura.
- L'algoritmo procede iterativamente, visitando successivamente alcuni tra i vertici del poliedro che definisce l'insieme delle soluzioni ammissibili.
- Dato un vertice x^* , si cerca prima di tutto di determinare se tale vertice sia o meno ad una soluzione ottima, cercando di determinare se esiste una soluzione y^* per il duale con lo stesso valore della funzione obiettivo.
- Nel caso in cui x^* non sia ottima, si cerca di spostarsi in un altro vertice, seguendo una direzione di crescita, che sia anche ammissibile.

- Se è possibile spostarsi indefinitamente lungo questa direzione di crescita, allora il problema è illimitato, altrimenti si incontra un altro vertice, e ci si sposta.

SIMPLESSOPRIMALE(A,b,c,B)

1. $N \leftarrow \{1, \dots, m\} - B;$
2. $\bar{x} \leftarrow A_B^{-1}b_B;$
3. $\bar{y}_B \leftarrow cA_B^{-1};$
4. $\bar{y}_N \leftarrow 0;$
5. Se $\bar{y}_B \geq 0$, allora termina con successo e restituisci \bar{x} e \bar{y} ;
6. $h \leftarrow \min\{i \in B \mid \bar{y}_i < 0\};$
7. Sia ξ la colonna di indice h in $-(A_B^{-1});$
8. Se $A_N\xi \leq 0$, allora termina e restituisci ξ : il problema è illimitato;
9. $k \leftarrow \arg \min\{\frac{b_i - A_i\bar{x}}{A_i\xi} \mid A_i\xi > 0 \wedge i \in N\};$
10. $B \leftarrow B \cup \{k\} - \{h\};$
11. Torna al punto 1.

Correttezza del simplesso

- L'algoritmo lavora mantenendo i seguenti tre invarianti:
 - B è una **base ammissibile**
 - x^* è **soluzione ammissibile** per il problema primale
$$\max \{ cx \mid Ax \leq b \};$$
mentre $y^*A = c$.
- Questo significa, tra l'altro, che x^* è sempre un vertice.
- Osserviamo che per y^* la condizione $y^*A = c$ vale per come y_B^* e y_N^* vengono inizializzati.
 - Di conseguenza, y^* è soluzione per il duale
$$\max\{yb \mid yA = c, y \geq 0\}$$
sse $y_B^* \geq 0$
- Se, quindi $y_B^* \geq 0$, allora l'algoritmo correttamente termina, restituendo x^* e y^* che sono soluzioni ottime per il primale e per il duale
- Se, invece, c'è un elemento di y^* strettamente negativo, allora non vale l'ottimalità. Cerchiamo quindi una direzione ammissibile e di crescita per x^* .
 - ξ è sempre **direzione di crescita**, perché
$$c\xi = c(-A^{-1}B u_h) = -(cA^{-1}B)u_h = -y_h u_h = -y_h > 0$$
dove u_h è un vettore ovunque nullo, tranne nella componente corrispondente a i , in cui vale 1.

- ξ , però, potrebbe non essere direzione ammissibile, e soprattutto, non sappiamo quale sia il “primo” vertice lungo ξ .
- Il vettore $A_B \xi$ è una delle colonne della matrice identica, cambiata di segno, e quindi $A_B \xi \leq 0$
- Se $i \in N$ e $A_i \xi \leq 0$, allora $x^*(\lambda)$ soddisfa l’i-esimo vincolo per ogni valore non-negativo di λ .
- Se $i \in N$ e $A_i \xi > 0$, allora

$$A_i x^*(\lambda) = A_i x^* + \lambda A_i \xi \leq b_i \Leftrightarrow \lambda \leq (b_i - A_i x^*) / A_i \xi$$
- Scegliamo l’indice i che rende tale λ minimo. Chiamiamolo k .
- Sia λ^* il valore $\min\{\lambda_i \mid i \in N\}$.
- 1. Se $\lambda^* = +\infty$, ossia se $A_N \xi < 0$, allora il problema è illimitato.
 - Questo caso è gestito dalla riga 8. dell’Algoritmo.
- 2. Se $0 < \lambda^* < +\infty$, allora $x(\lambda)$ è ammissibile per ogni $\lambda \in [0, \lambda^*]$ e non ammissibile altrimenti. Possiamo quindi spostarci da B a $B \cup \{k\} - \{h\}$, che corrisponde ad un altro vertice.
 - Questo caso è gestito dalle righe 9.-10. dell’Algoritmo.
- 3. Se $\lambda^* = 0$, allora la direzione non è ammissibile, ma possiamo comunque effettuare un cambio di base verso $B \cup \{k\} - \{h\}$, che ci fa restare sullo stesso vertice.
 - Questo caso è gestito dalle righe 9.-10. dell’Algoritmo.

Complessità del simplesso

- Si può dimostrare che ogni base ammissibile viene trattata al più una volta durante l’esecuzione dell’Algoritmo.
- Di conseguenza, vi saranno al più $\binom{m}{n}$ iterazioni, ovvero un numero che può divenire esponenziale in n .
- Detto questo:
 - Da un punto di vista teorico, la complessità dell’Algoritmo nel caso medio è polinomiale.
 - Da un punto di vista pratico, si osserva come il Simplex sia l’algoritmo più efficiente, e che si comporti meglio di altri algoritmi (alcuni dei quali si possono dimostrare essere polinomiali in tempo).