

Marco Casu

🌀 Ottimizzazione 🌀



SAPIENZA  
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica  
Dipartimento di Informatica

Questo documento è distribuito sotto la licenza [GNU](#), è un resoconto degli appunti (eventualmente integrati con libri di testo) tratti dalle lezioni del corso di Ottimizzazione per la laurea triennale in Informatica. Se dovessi notare errori, ti prego di segnalarmeli.



# INDICE

<b>1</b>	<b>Flussi nei Grafi</b>	<b>3</b>
1.1	Definizione e Grafo Residuo . . . . .	3
1.2	Tagli $s - t$ . . . . .	5
1.3	Percorso Minimo nell'Aumento del Flusso . . . . .	7
1.4	Cammini Edge-Disjoint in un Grafo . . . . .	9
<b>2</b>	<b>Programmazione Lineare</b>	<b>10</b>
2.1	Insiemi Convessi . . . . .	10
2.2	Applicazioni della Programmazione Lineare . . . . .	13
2.3	Il Metodo del Simplexso . . . . .	15
2.3.1	Soluzioni Ammissibili Basiche . . . . .	16
2.3.2	Ricerca sul Poliedro . . . . .	20
2.3.3	La Procedura di Risoluzione . . . . .	25
2.4	Il Tableau del Simplexso . . . . .	28
2.5	Soluzioni Degenere . . . . .	30

## CAPITOLO

# 1

## FLUSSI NEI GRAFI

### 1.1 Definizione e Grafo Residuo

**Definizione 1** Una **network** o **rete**  $G = (V, E, c, s, t)$  è un particolare grafo diretto, in cui  $V$  ed  $E$  sono i vertici e gli archi, tali per cui è soddisfatta la condizione

$$\forall (u, v) \in E(G), \quad \exists (v, u) \in E(G)$$

$c : E(G) \rightarrow \mathbb{R}^+$  è una funzione detta **capacità**,  $s$  e  $t$  sono due particolari vertici in  $V(G)$  denominati **source** e **sink**.

**Definizione 2** Data una network  $G = (V, E, c, s, t)$ , un **flusso** per  $G$  è una funzione  $f : E(G) \rightarrow \mathbb{R}$  tale per cui valgono le seguenti

1. **skew-simmetria**:  $f(u, v) = -f(v, u), \quad \forall (u, v) \in E(G)$
2. **capacità rispettata**:  $f(u, v) \leq c(u, v), \quad \forall (u, v) \in E(G)$
3. **conservatività del flusso**:  $\sum_{(u, v) \in E(G)} f(u, v) = 0, \quad \forall v \in V(G) \setminus \{s, t\}$

Denominiamo flusso uscente dal vertice  $v$  la somma del flusso (positivo) valutato su tutti gli archi che hanno  $v$  come primo membro (che collegano  $v$  ad un'altro vertice). Analogamente (ma in maniera opposta) si definisce il flusso entrante. Dato un flusso  $f$  per una network  $G$  si definisce il **valore del flusso** la somma del flusso uscente da  $s$

$$\text{val}(f) = \sum_{(s, u) \in E(G)} f(s, u)$$

La terza proprietà, di conservazione del flusso, asserisce che il flusso uscente da un nodo deve essere identico al flusso entrante, sia  $x$  un vertice fissato in  $V(G)$

$$\sum_{\substack{(u, x) \in E(G) \\ f(u, x) > 0}} f(u, x) = - \left( \sum_{\substack{(x, u) \in E(G) \\ f(x, u) < 0}} f(x, u) \right)$$

**Definizione 3** Sia  $G = (V, E, c, s, t)$  una network e  $f$  un flusso per  $G$ , il **grafo residuo** è il grafo diretto  $G'$  definito come segue

- $\forall v \in V(G), v \in V(G')$
- $(u, v) \in E(G) \wedge f(u, v) < c(u, v) \implies (u, v) \in E(G')$

Inoltre è definita una funzione  $r : E(G') \rightarrow \mathbb{R}^+$  detta **capacità residua** definita come segue

$$r(u, v) = c(u, v) - f(u, v)$$

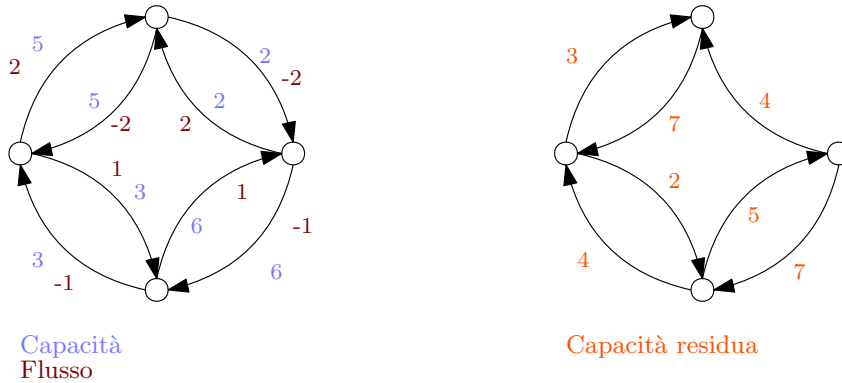


Figura 1.1: Capacità residua del flusso (evidenziato in rosso)

Si assuma che esiste un cammino  $P$  in  $G'$  da  $s$  a  $t$ , si consideri il residuo minimo valutato sugli archi contenuti nel cammino

$$\alpha = \min_{(u,v) \in E(P)} r(u, v)$$

Si definisce una funzione  $f' : E(G) \rightarrow \mathbb{R}$  come segue

$$f'(u, v) = \begin{cases} f(u, v) + \alpha & \text{se } (u, v) \in E(P) \\ f(u, v) - \alpha & \text{se } (v, u) \in E(P) \\ f(u, v) & \text{altrimenti} \end{cases}$$

**Proposizione 1**  $f'$  è un flusso per  $G$ .

*Dimostrazione* : Sia  $(u, v)$  un arco in  $G$ , se  $(u, v) \notin E(P)$ , allora  $f'(u, v) = f(u, v)$  e conseguentemente  $f'(v, u) = f(v, u)$ , quindi la proprietà di skew simmetria è preservata. Differentemente, se  $(u, v) \in E(P)$  si avrebbe che  $f'(u, v) = f(u, v) + \alpha$  e  $f'(v, u) = f(v, u) - \alpha = -f(u, v) - \alpha = -(f(u, v) + \alpha)$ , quindi il nuovo flusso rispetta la proprietà di skew-simmetria.

Per ogni arco  $(u, v) \in E(P)$  si ha che  $f'(u, v) = f(u, v) + \alpha$ ,  $\alpha$  è (per definizione) minore o uguale a  $r(u, v)$  quindi

$$f'(u, v) \leq f(u, v) + r(u, v)$$

Ma essendo che  $f(u, v) + r(u, v) = c(u, v)$ ,  $f'$  rispetta la capacità.

Se  $x \notin V(P)$  si avrebbe che  $f'(x, u) = f(x, u)$  per ogni  $u$  adiacente ad  $x$ , allora

$$\sum_{(x,u) \in E(G)} f(x, u) = 0$$

Assumendo che  $x \in V(P)$ , vi è un arco uscente da  $x$  il cui flusso è aumentato di  $\alpha$ , vi è quindi (per definizione di  $f'$ ) un'arco entrante in  $x$  il cui flusso è diminuito di  $\alpha$ , quindi è ancora vero che

$$\sum_{\substack{(u,x) \in E(G) \\ f'(u,x) > 0}} f'(u, x) = - \left( \sum_{\substack{(x,u) \in E(G) \\ f'(x,u) < 0}} f'(x, u) \right)$$

la proprietà di conservazione del flusso è rispettata. ■

Il valore del nuovo flusso è uguale al valore del flusso di partenza aumentato di  $\alpha$

$$\text{val}(f') = \text{val}(f) + \alpha$$

**Algorithm 1** Ford–Fulkerson**Require:** network  $G = (V, E, c, s, t)$ si definisce un flusso  $f$  tale che  $f(u, v) = 0, \forall (u, v) \in E(G)$ si definisce il grafo residuo  $G'$  dato il flusso  $f$ **while** Esiste un cammino  $P$  in  $G'$  da  $s$  a  $t$  **do**    si definisce la funzione delle capacità residue  $r : E(G') \rightarrow \mathbb{R}$      $\alpha = \min_{(u,v) \in E(P)} r(u, v)$     Si definisce un flusso  $f' = f$     **for**  $(u, v) \in E(P)$  **do**         $f'(u, v) = f(u, v) + \alpha$          $f'(v, u) = f(v, u) - \alpha$     **end for****end while**

Dato che un singolo arco  $(s, u)$  per qualche  $u$  è necessariamente presente nel cammino  $P$  da  $s$  a  $t$ , ed il valore di  $f'$  su  $(s, u)$  è stato aumentato di  $\alpha$ . La proposizione 1 delinea una procedura per la ricerca di un flusso ottimale (di valore massimo) per una network. Alla fine dell'esecuzione, il flusso  $f'$  sarà ottimale per la network data.

**Osservazione 1** Se le capacità della network sono numeri interi, l'algoritmo termina. Se invece le capacità sono numeri reali, l'algoritmo potrebbe non terminare.

## 1.2 Tagli $s - t$

Data una network  $G = (V, E, c, s, t)$ , ed un flusso  $f$  per  $G$ , si consideri un'insieme  $\mathcal{U} \subset V(G)$  tale che

- $s \in \mathcal{U}$
- $t \notin \mathcal{U}$

Tale insieme è detto **insieme di taglio**, si consideri ora il flusso uscente dai vertici presenti in  $\mathcal{U}$

$$\sum_{\substack{(u,x) \in E(G) \\ \text{t.c. } u \in \mathcal{U}}} f(u, x)$$

Per la proprietà di conservazione del flusso si ha che il flusso uscente da ogni vertice diverso da  $s$  è nullo, ed il flusso uscente dal vertice  $s$  è il valore del flusso.

$$\sum_{\substack{(u,x) \in E(G) \\ \text{t.c. } u \in \mathcal{U}}} f(u, x) = \sum_{(s,x) \in E(G)} f(s, x) = \text{val}(f)$$

La sommatoria a sinistra può essere riscritta come la somma del flusso uscente dai vertici in  $\mathcal{U}$  verso i vertici in  $\mathcal{U}$ , e del flusso uscente dai vertici in  $\mathcal{U}$  verso i vertici che non sono contenuti in  $\mathcal{U}$

$$\sum_{\substack{(u,x) \in E(G) \\ \text{t.c. } u \in \mathcal{U}}} f(u, x) = \sum_{\substack{(u,x) \in E(G) \\ \text{t.c. } u, x \in \mathcal{U}}} f(u, x) + \sum_{\substack{(u,x) \in E(G) \\ \text{t.c. } u \in \mathcal{U} \\ x \notin \mathcal{U}}} f(u, x)$$

Per la proprietà di skew-simmetria il flusso uscente dai vertici in  $\mathcal{U}$  verso i vertici in  $\mathcal{U}$  è nullo

$$\sum_{\substack{(u,x) \in E(G) \\ \text{t.c. } u \in \mathcal{U}}} f(u, x) = \sum_{\substack{(u,x) \in E(G) \\ \text{t.c. } u \in \mathcal{U} \\ x \notin \mathcal{U}}} f(u, x)$$

**Conclusione** : Il valore di  $f$  è uguale alla somma dei flussi uscenti dai vertici in  $\mathcal{U}$  verso i vertici non contenuti in  $\mathcal{U}$ . Questa proprietà è invariante rispetto la scelta di  $\mathcal{U}$ , a patto che rispetti le proprietà inizialmente elencate (deve contenere  $s$  ma non  $t$ ).

**Definizione 4** si definisce **capacità di taglio** la somma delle capacità degli archi che collegano i vertici in  $\mathcal{U}$  ai vertici in  $V(G) \setminus \mathcal{U}$

$$c_t = \sum_{\substack{(u,x) \in E(G), \\ u \in \mathcal{U}, \\ x \notin \mathcal{U}}} c(u,x)$$



**Osservazione 2** il valore massimale del flusso è limitato dalla capacità di taglio

$$\text{val}(f) \leq c_t$$

**Proposizione 2** Data una network  $G$ , se esiste un flusso  $f^*$  ed un'insieme di taglio  $\mathcal{U}$  tali che

$$\text{val}(f) = \sum_{\substack{(u,x) \in E(G), \\ u \in \mathcal{U}, \\ x \notin \mathcal{U}}} c(u,x)$$

ossia, il valore del flusso è identico alla capacità di taglio, allora  $f^*$  è un flusso ottimale.

L'algoritmo di Ford-Fulkerson restituisce un flusso ottimale  $f^*$ , da questo è possibile individuare l'insieme di taglio  $\mathcal{U}$  associato, in particolare, se  $G^*$  è il grafo residuo della network rispetto il flusso dato in output  $f^*$ , allora l'insieme di taglio sarà composto da tutti i nodi raggiungibili da  $s$  in  $G^*$ , chiaramente, fra questi non vi sarà  $t$ , data la definizione dell'algoritmo, che termina proprio quando non vi è un cammino da  $s$  a  $t$ .

Si consideri adesso una network  $G$ , di cui  $f^*$  è il flusso ottimale trovato tramite l'algoritmo 1. Sia  $\mathcal{U}$  l'insieme di taglio dato dai nodi raggiungibili da  $s$  nel grafo residuo  $G^*$ .

**Osservazione 3** Per ogni arco  $(x,y) \in E(G)$  con  $x \in \mathcal{U}$  e  $y \notin \mathcal{U}$ , si avrà che

$$f^*(x,y) = c(x,y)$$

Il valore del flusso è uguale alla somma delle capacità degli archi che collegano i vertici in  $\mathcal{U}$  a quelli fuori da  $\mathcal{U}$

$$\sum_{\substack{(x,y) \in E(G) \\ x \in \mathcal{U} \\ y \notin \mathcal{U}}} f^*(x,y) = \sum_{\substack{(x,y) \in E(G) \\ x \in \mathcal{U} \\ y \notin \mathcal{U}}} c(x,y) = \text{val}(f^*)$$

La proposizione 2 non implica che non ci possa essere una network il cui flusso ottimale a valore strettamente minore della capacità di taglio di uno specifico insieme  $\mathcal{U}$ , si consideri l'immagine in figura 1.2, in cui è applicata la notazione sugli archi *capacità/flusso*, la capacità di taglio è data dalla somma delle capacità sugli archi evidenziati, ed è uguale a 4, nonostante questo, il flusso ottimale per la network in questione ha valore 1.

Nonostante ciò, esiste sempre un insieme  $\mathcal{U}$  contenente  $s$  e non  $t$  la cui capacità di taglio è uguale al valore del flusso ottimale per la network data, tale insieme può essere trovato adoperando l'algoritmo di Ford-Fulkerson nella procedura precedentemente elencata.

**Osservazione 4** L'algoritmo di Ford-Fulkerson, termina sempre se le capacità della network sono numeri in  $\mathbb{Q}$ .

*Dimostrazione* : Se le capacità  $c_i$  sono numeri razionali allora esiste un numero naturale  $N \in \mathbb{N}$  tale che ogni capacità è della forma  $c_i = \frac{a_i}{N}$ , ad ogni iterazione dell'algoritmo il valore del flusso aumenta di almeno  $\frac{1}{N}$ , quindi in un numero finito di passi raggiungerà il valore ottimale. ■

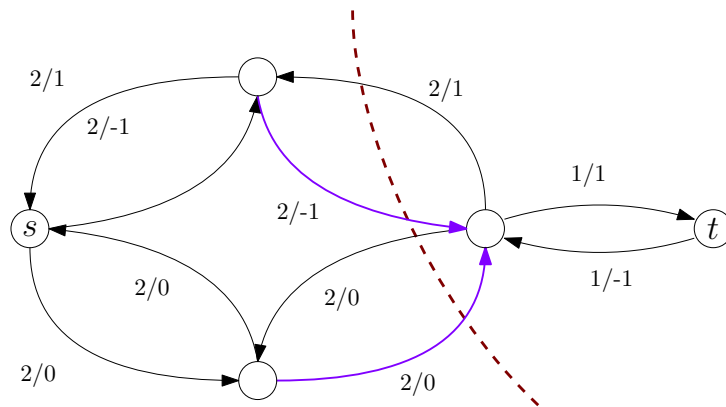


Figura 1.2: network con taglio sui vertici

### 1.3 Percorso Minimo nell'Aumento del Flusso

Durante la computazione dell'algoritmo di Ford-Fulkerson, viene scelto un qualsiasi percorso che connetta  $s$  a  $t$  nel grafo residuo, tale scelta comporta un aumento del valore del flusso, ma una scelta differente di percorso potrebbe far sì che l'aumento in quella iterazione sia maggiore, e che il numero finale di iterazioni per trovare il flusso ottimale sia minore. Il seguente esempio mostra l'inefficienza dell'algoritmo 1, si consideri la network in figura 1.3 (alcuni archi sono stati omessi). Il flusso massimale ha valore  $2M$ ,

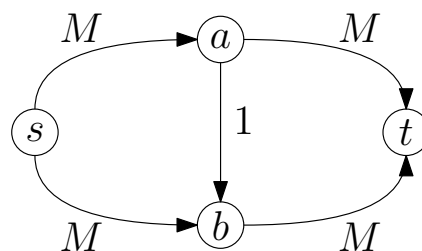


Figura 1.3: Sugli archi sono indicate le capacità

nonostante ciò, se ad ogni iterazione dell'algoritmo venisse selezionato il percorso  $s \rightarrow a \rightarrow b \rightarrow t$ , allora l'aumento del valore sarebbe uguale ad uno, e sarebbero necessarie  $2M$  iterazioni, diversamente, la scelta del percorso  $s \rightarrow a \rightarrow t$  implicherebbe già solo alla prima iterazione un'aumento pari ad  $M$ .

La complessità computazionale in questo caso dipende linearmente da  $M$ , tale valore è però codificato in binario (occupando  $\log M$  spazio), quindi l'algoritmo di Ford-Fulkerson è esponenziale nelle dimensioni dell'input. È possibile considerare una rivisitazione dell'algoritmo 1, in cui ad ogni iterazione viene selezionato il percorso più breve (minor numero di archi) da  $s$  a  $t$  nel grafo residuo. Tale algoritmo rivisitato è noto con il nome di **Edmonds-Karp**.

---

#### Algorithm 2 Edmonds-Karp

---

**Require:** network  $G = (V, E, c, s, t)$

si definisce un flusso  $f$  tale che  $f(u, v) = 0, \forall (u, v) \in E(G)$

si definisce il grafo residuo  $G'$  dato il flusso  $f$

**while** Esiste un cammino  $P$  in  $G'$  da  $s$  a  $t$  **do**

$P$  = cammino più breve da  $s$  a  $t$  in  $G'$

    si definisce la funzione delle capacità residue  $r : E(G') \rightarrow \mathbb{R}$

$\alpha = \min_{(u,v) \in E(P)} r(u, v)$

    Si definisce un flusso  $f' = f$

**for**  $(u, v) \in E(P)$  **do**

$f'(u, v) = f(u, v) + \alpha$

$f'(v, u) = f(v, u) - \alpha$

**end for**

**end while**

---





**Osservazione 5** Se  $G$  è un grafo diretto e  $P$  è il percorso più breve fra due vertici  $x$  ed  $y$ , allora  $\forall z \in V(P)$ , si ha che il sotto cammino  $x \rightarrow z$  in  $P$  è anch'esso un percorso più breve.

**Proposizione 3** Sia  $G = (V, E, c, s, t)$  una network. Sia  $G_i$  il grafo residuo all' $i$ -esima iterazione dell'algoritmo 2, e  $G_{i'}$  il grafo residuo all' $i'$ -esima iterazione, con  $i' > i$ , allora

$$\text{dist}_{G_i}(s, u) \leq \text{dist}_{G_{i'}}(s, u) \quad (1.1)$$

La distanza dal vertice source  $s$  rispetto ogni altro vertice aumenta in maniera monotona ad ogni passo dell'algoritmo.

*Dimostrazione* : Supponiamo che esiste un nodo  $v \in G$  tale che  $\text{dist}_{G_i}(s, v) > \text{dist}_{G_{i'}}(s, v)$ , si assume inoltre che la distanza  $\text{dist}_{G_{i'}}(s, v)$  sia la più piccola possibile ( $v$  è il nodo più vicino ad  $s$  in  $G_{i'}$ ). Sia  $w$  il penultimo vertice del cammino  $P' = u_1, u_2 \dots u_k$  in  $G_{i'}$ , con  $u_1 = s$  e  $u_k = v$ . Ne segue che



$$\text{dist}_{G_i}(s, v) > \text{dist}_{G_{i'}}(s, v) = \text{dist}_{G_{i'}}(s, w) + 1 \geq \text{dist}_{G_i}(s, w) + 1 \quad (1.2)$$

**Nota** : nella dimostrazione si sta assumendo che la proposizione non sia valida per il nodo  $v$ , ma che sia valida per il nodo  $w$ , da qui è verificata la disuguaglianza a destra nell'equazione 1.2.

Ciò implica che l'arco  $(w, v)$  è presente in  $G_{i'}$  ma non in  $G_i$ , se così non fosse sarebbe vero che  $\text{dist}_{G_{i'}}(s, w) \geq \text{dist}_{G_i}(s, w) + 1$ , e quindi  $w = u_i$  e  $v = u_{i+1}$  per qualche  $i$ , ma questa è una contraddizione dato che  $v$  segue  $w$  nel cammino  $P'$ , quindi l'asserto è verificato. ■

**Teorema 1** Nell'algoritmo di Edmonds-Karp, applicato su una network  $G$ , il numero totale di incrementi del valore del flusso è al più  $n \cdot m$ , con  $n = |V(G)|$  e  $m = |E(G)|$ . Tale affermazione è valida anche se le capacità sugli archi sono numeri reali.

*Dimostrazione* : Sia  $G_i$  il grafo residuo nell' $i$ -esima iterazione dell'algoritmo 2, analogamente, sia  $f_i$  il flusso valutato anch'esso durante l' $i$ -esima iterazione. Chiaramente  $G_0 = G$  e  $f_0(e) = 0, \forall e$ .

**Definizione 5** Durante l'esecuzione dell'algoritmo 2, un'arco  $(u, v)$  è detto **critico** in  $i$  se

- $(u, v) \in G_i$
- $(u, v) \notin G_{i+1}$

Se  $P_i$  è il percorso minimo da  $s$  a  $t$  considerato nell' $i$ -esima iterazione, e  $(u, v)$  è critico in  $i$ , per definizione dell'algoritmo si ha che  $(u, v) \in E(P_i)$ .

**Lemma 1** Sia  $(u, v)$  un'arco di una network  $G$ , durante l'esecuzione dell'algoritmo 2, l'arco  $(u, v)$  può essere considerato critico al più  $\frac{n}{2}$  volte.

*Dimostrazione Lemma* : Siano

$$\pi(1) < \pi(2) < \dots < \pi(L)$$

gli indici delle iterazioni in cui  $(u, v)$  è critico, con  $L \leq \frac{n}{2}$ , chiaramente

$$(u, v) \in E(P_{\pi(i)})$$

per qualche  $1 \leq i \leq L$ . Chiaramente

$$\text{dist}_{G_{\pi(i)}}(s, v) = \text{dist}_{G_{\pi(i)}}(s, u) + 1$$

Se  $(u, v)$  è critico in  $\pi(i)$  ed in  $\pi(i+1)$ , allora deve esistere un iterazione  $i'$  compresa fra queste

$$\pi(i) < i' < \pi(i+1)$$

In cui l'arco  $(u, v)$  è stato re-inserito nel grafo residuo, quindi il flusso su  $(u, v)$  in tale iterazione è diminuito, necessariamente (per skew-simmetria) il flusso su  $(v, u)$  è aumentato, quindi quest'ultimo arco si trovava sul percorso da  $s$  a  $t$  nell'iterazione  $i'$ .

$$(v, u) \in E(P_{i'})$$

Date le precedenti osservazioni, si deducono le seguenti disuguaglianze

$$\text{dist}_{G_{i'}}(s, u) = \text{dist}_{G_{i'}}(s, v) + 1 \quad (1.3)$$

$$\text{dist}_{G_{i'}}(s, v) + 1 \geq \text{dist}_{G_{\pi(i)}}(s, v) + 1 \quad (1.4)$$

$$\text{dist}_{G_{\pi(i)}}(s, v) + 1 = \text{dist}_{G_{\pi(i)}}(s, u) + 2 \quad (1.5)$$

$$\Downarrow \quad (1.6)$$

$$\text{dist}_{G_{\pi(i+1)}}(s, u) \geq \text{dist}_{G_{\pi(i)}}(s, u) + 2 \quad (1.7)$$

Essendo che la distanza fra due vertici è limitata da  $n = |V(G)|$ , si ha che

$$\text{dist}_{G_{\pi(i)}}(s, u) \leq n - 1$$

- la distanza fra  $s$  ed  $u$  è al più  $n - 1$
- la distanza fra  $s$  ed  $u$  aumenta almeno di due in due iterazioni differenti in cui  $(u, v)$  è critico

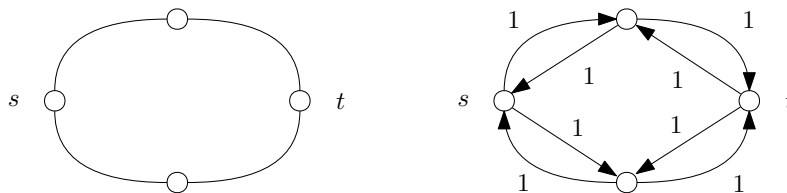
La conclusione è che non possono esistere più di  $\frac{n}{2}$  indici  $\pi(i)$  in cui  $(u, v)$  è critico.  $\square$

La dimostrazione del teorema 1 segue in maniera naturale, ad ogni iterazione un'arco è critico, essendo che ci sono  $m$  archi ed ognuno può essere critico al più  $\frac{n}{2}$  volte, il numero totale di aumenti del flusso è al più  $\frac{1}{2}nm$ .  $\blacksquare$

## 1.4 Cammini Edge-Disjoint in un Grafo

In questa sezione verrà esposta un'applicazione dell'algoritmo di ricerca del flusso massimo. Sia  $G$  un grafo non diretto, si definisce la *network associata* a  $G$ , il grafo diretto  $\vec{G}$  tale che

- $u \in V(G) \implies u \in V(\vec{G})$
- $(u, v) \in E(G) \implies \begin{cases} (u, v) \in E(\vec{G}) \\ (v, u) \in E(\vec{G}) \end{cases}$
- $\forall (u, v) \in \vec{G} \quad , c(u, v) = 1$



Si può anche risalire in maniera naturale ad un grafo non diretto associato ad una network.

**Definizione 6** Dato un grafo non diretto  $G$  e due nodi  $s, t$ , un'insieme di cammini da  $s$  a  $t$  è **edge-disjoint** se non condividono alcun arco.

Verrà mostrato come, dato un grafo  $G$ , il numero massimo di cammini edge-disjoint è uguale al valore del flusso ottimale nella network associata.

**Definizione 7** Dato un flusso  $f$  per una network  $\vec{G}$ , si definisce **supporto del flusso** l'insieme

$$W = \{(u, v) \in E(\vec{G}) \text{ t.c. } f(u, v) \neq 0\}$$

**Proposizione 4** Se nel supporto  $W$  di un flusso  $f$  per una network  $\vec{G}$  gli archi compongono  $k$  cammini diretti da  $s$  a  $t$ , allora ci sono  $k$  cammini edge-disjoint da  $s$  a  $t$  nel grafo non diretto associato.

*Dimostrazione :* **TODO : continuare**

# PROGRAMMAZIONE LINEARE

## 2.1 Insiemi Convessi

La programmazione lineare consiste nella ricerca di un vettore (ingresso di una funzione lineare) in cui tale funzione assume il valore massimo, all'interno di un dominio definito da un'insieme di vincoli lineari. La funzione da massimizzare è detta **funzione obiettivo**, un'esempio di programma lineare può essere il seguente

$$x_1 + x_2$$

soggetto ai vincoli

$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_2 - x_1 &\leq 1 \\ x_1 + 6x_2 &\leq 15 \\ 4x_1 - x_2 &\leq 10 \end{aligned}$$

Un punto è **ammissibile** se soddisfa tutti i vincoli lineari. L'insieme di punti ammissibili si può rappresentare su un piano in questo caso, essendo un sotto-insieme di  $\mathbb{R}^2$ . La funzione obiettivo essendo lineare



Figura 2.1: Insieme dei punti ammissibili

si può rappresentare come prodotto scalare fra due vettori  $\mathbf{c}$  e  $\mathbf{x}$

$$\mathbf{c}^T \mathbf{x} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 + x_2$$

Può risultare utile rappresentare sul piano anche il vettore  $\mathbf{c}$  e la retta equivalente al sottospazio

$$\text{span}(\mathbf{c}) = \{\alpha\mathbf{c} \mid \alpha \in \mathbb{R}\}$$

Si consideri una retta  $y'$  perpendicolare alla retta definita da  $\text{span}(\mathbf{c})$ , i punti di  $y'$  che intersecano



l'insieme delle soluzioni ammissibili condividono la stessa immagine se valutati sulla funzione obiettivo. Un'interpretazione geometrica del problema può essere la seguente

Massimizzare la funzione obiettivo equivale a trovare il massimo  $\beta$  tale che l'iperpiano definito da  $\mathbf{c}^T \mathbf{x} = \beta$  perpendicolare alla retta  $\alpha\mathbf{c}$  interseca l'insieme dei punti ammissibili.

Si definisce **soluzione ottimale** ogni soluzione ammissibile che massimizza la funzione obiettivo. Il

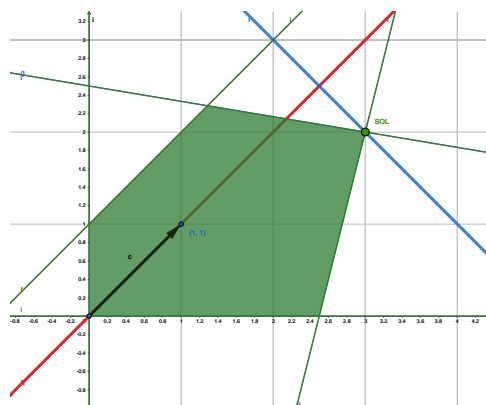


Figura 2.2: Il punto  $(3, 2)$  è una soluzione ottimale per il programma lineare

numero di soluzioni di un programma lineare può variare fra i seguenti casi

1. Vi è un'unica soluzione ottimale.
2. Vi sono infinite soluzioni ottimali.
3. Non ci sono soluzioni ammissibili, nessun punto soddisfa tutti i vincoli lineari.
4. Non ci sono soluzioni ottimali perché il problema non è limitato, ciò avviene se l'insieme dei punti ammissibili non è chiuso.

Generalmente la funzione obiettivo  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  è del tipo

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

e gli  $m$  vincoli sono della forma

$$\begin{aligned} a_{11}x_1 + a_{21}x_2 + \cdots + a_{n1}x_n &\leq b_1 \\ a_{12}x_1 + a_{22}x_2 + \cdots + a_{n2}x_n &\leq b_2 \\ &\vdots \\ a_{1m}x_1 + a_{2m}x_2 + \cdots + a_{nm}x_n &\leq b_m \end{aligned}$$

**Osservazione 6** Massimizzare  $\mathbf{c}^T \mathbf{x}$  equivale a minimizzare  $-\mathbf{c}^T \mathbf{x}$ .

Si può quindi assumere che un generico problema di programmazione lineare riguardi la massimizzazione di una funzione del tipo  $\mathbf{c}^T \mathbf{x}$  per un fissato  $\mathbf{c} \in \mathbb{R}^n$ . Inoltre, ogni vincolo del tipo

$$a_{1i}x_1 + a_{2i}x_2 + \cdots + a_{ni}x_n \leq b_i \quad (2.1)$$

è soddisfatto dagli stessi punti che soddisfano

$$-a_{1i}x_1 - a_{2i}x_2 - \cdots - a_{ni}x_n \geq b_i$$

Quindi si può assumere che ogni vincolo sia scritto nella forma 2.1. Inoltre ogni vincolo di uguaglianza equivale a due disuguaglianze. Date le precedenti osservazioni, si può definire un'insieme di  $m$  vincoli in maniera compatta tramite una matrice  $m \times n$  ed un vettore  $\mathbf{b} \in \mathbb{R}^m$ .

$$A\mathbf{x} \leq \mathbf{b}$$

Inoltre ogni vincolo di positività del tipo  $x_i \geq 0$  equivale ad il vincolo  $-x_i \leq 0$ . Se in un programma lineare una variabile  $x_i$  può assumere qualsiasi valore in  $\mathbb{R}$ , si può sostituire con la differenza di due nuove variabili

$$x_i = z_i - z'_i$$

ed imporre i vincoli

$$z_i, z'_i \geq 0$$

in tal modo è possibile, per ogni variabile del programma lineare, imporre la positività. Ciò è utile per definire una *forma standard* per un programma lineare.

**Definizione 8** Un programma lineare in **forma standard** è un problema di ottimizzazione del tipo

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Dove

$$\begin{aligned} \mathbf{x}, \mathbf{c} &\in \mathbb{R}^n \\ A &\in \text{Mat}(m \times n) \\ \mathbf{b} &\in \mathbb{R}^m \end{aligned}$$

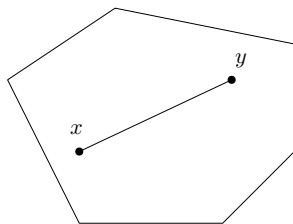
L'esistenza di soluzioni ammissibili dipende esclusivamente dalla matrice  $A$ . È impossibile che un programma lineare abbia un numero finito di soluzioni diverso da 0 e 1. In seguito verrà dimostrato che se un programma lineare ha due punti ammissibili che sono soluzione, allora ha infinite soluzioni.

**Definizione 9** Dati due punti  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , si definisce **segmento di linea** fra i punti l'insieme

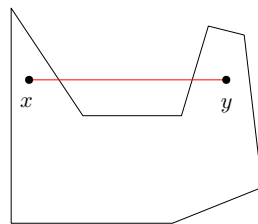
$$\{\alpha \mathbf{x} + (1 - \alpha) \mathbf{y} \text{ t.c. } \alpha \in [0, 1]\}$$

**Definizione 10** Un sotto-insieme  $X \subseteq \mathbb{R}^n$  è **convesso** se la seguente è verificata

$$\forall \mathbf{x}, \mathbf{y} \in X, \quad \{\alpha \mathbf{x} + (1 - \alpha) \mathbf{y} \text{ t.c. } \alpha \in [0, 1]\} \subseteq X$$



sotto-insieme di  $\mathbb{R}^2$  convesso



sotto-insieme di  $\mathbb{R}^2$  non convesso

**Proposizione 5** L'insieme dei punti ammissibili di un programma lineare è convesso.



*Dimostrazione* : Siano  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  due punti ammissibili, sia  $\alpha \in [0, 1]$  fissato, si considera il punto (appartenente al segmento)

$$\alpha \mathbf{x} + (1 - \alpha) \mathbf{y}$$

si ha che

$$A(\alpha \mathbf{x} + (1 - \alpha) \mathbf{y}) = \quad (2.2)$$

$$\alpha A\mathbf{x} + (1 - \alpha) A\mathbf{y} = \quad (2.3)$$

$$\alpha \mathbf{b} + (1 - \alpha) \mathbf{b} = \mathbf{b} \quad (2.4)$$

quindi Il punto soddisfa gli  $m$  vincoli

$$A(\alpha \mathbf{x} + (1 - \alpha) \mathbf{y}) \leq \mathbf{b}$$

Inoltre essendo che

$$\alpha \geq 0, \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}$$

si ha che

$$\alpha \mathbf{x} + (1 - \alpha) \mathbf{y} \geq \mathbf{0}$$

Quindi  $\alpha \mathbf{x} + (1 - \alpha) \mathbf{y}$  soddisfa tutti i vincoli del programma lineare, ed appartiene quindi ai punti ammissibili. ■

**Proposizione 6** *Se per un programma lineare esistono due soluzioni ottimali distinte  $\mathbf{x}^*, \mathbf{y}^* \in \mathbb{R}^n$ , allora tale programma lineare ha infinite soluzioni ottimali.*

*Dimostrazione* : Sia  $\mathbf{z}$  un punto sul segmento delineato da  $\mathbf{x}^*, \mathbf{y}^*$

$$\mathbf{z} = \alpha \mathbf{x}^* + (1 - \alpha) \mathbf{y}^* \quad \text{per qualche } \alpha \in [0, 1]$$

si ha che

$$\mathbf{c}^T \mathbf{z} = \quad (2.5)$$

$$\mathbf{c}^T (\alpha \mathbf{x}^* + (1 - \alpha) \mathbf{y}^*) = \quad (2.6)$$

$$\alpha \mathbf{c}^T \mathbf{x}^* + (1 - \alpha) \mathbf{c}^T \mathbf{y}^* \quad (2.7)$$

Essendo che  $\mathbf{x}^*$  e  $\mathbf{y}^*$  sono entrambe soluzioni ottimali, si ha che  $\mathbf{c}^T \mathbf{x}^* = \mathbf{c}^T \mathbf{y}^*$

$$\alpha \mathbf{c}^T \mathbf{x}^* + (1 - \alpha) \mathbf{c}^T \mathbf{y}^* = \quad (2.8)$$

$$\alpha \mathbf{c}^T \mathbf{x}^* + (1 - \alpha) \mathbf{c}^T \mathbf{x}^* = \mathbf{c}^T \mathbf{x}^* \quad (2.9)$$

Quindi anche  $\mathbf{z}$  è soluzione, essendo che quest'ultimo è un generico punto sul segmento, tutti i punti del segmento sono soluzioni. ■

## 2.2 Applicazioni della Programmazione Lineare

Un classico esempio di applicazione riguarda la scelta di una dieta che soddisfi dei vincoli nutrizionali minimizzando il costo degli alimenti. Vi è un'insieme di  $n$  alimenti  $x_1, \dots, x_n$  ed ognuno di questi ha un costo  $c_i$ , si vuole minimizzare

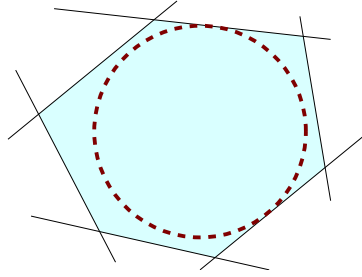
$$\sum_{i=1}^n c_i x_i$$

tenendo conto dei vincoli nutrizionali del tipo

$$a_{1i} x_1 + a_{2i} x_2 + \dots + a_{ni} x_n \geq b_i$$

Un'altro esempio riguarda la ricerca di un flusso ottimale per una network  $G = (V, E, c, s, t)$ , per ogni arco  $(i, j)$  vi sarà una variabile  $x_{ij}$  che rappresenta il flusso su tale arco, si vuole massimizzare la somma del flusso uscente dal vertice source  $s$

$$\sum_{(s,j) \in E(G)} x_{sj}$$



I vincoli riguardano la skew simmetria, la conservazione del flusso e il vincolo delle capacità, ad esempio:

$$\begin{aligned} x_{ij} &\leq c_{ij} \\ x_{ij} &= -x_{ji} \end{aligned}$$

Un'esempio interessante di problema che si riconduce alla programmazione lineare riguarda la ricerca del cerchio di raggio massimo che può essere inscritto in un poligono. Si assume che i lati del poligono possono essere descritti da un'insieme di rette del tipo  $y = mx + b$  con  $m \neq 0$ .

Si consideri una delle rette  $y = mx + b$ , si vuole trovare una retta perpendicolare a questa passante per un fissato punto  $(x_0, y_0)$ . Questa ha equazione

$$y = \frac{y_0 - x}{m} + y_0$$

Il punto di intersezione è  $P = (x', y')$  dove

$$x' = \frac{x_0 + my_0}{m^2 + 1} - mb \quad (2.10)$$

$$y' = m \frac{x_0 + my_0 - mb}{m^2 + 1} + b \quad (2.11)$$

La distanza fra  $P$  e  $(x_0, y_0)$ , applicando la norma euclidea, risulta essere

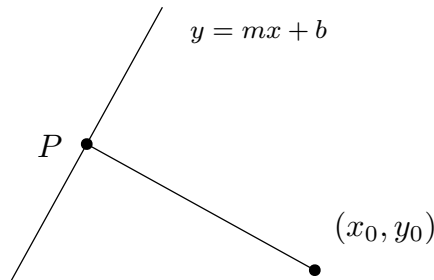


Figura 2.3: retta perpendicolare passante per  $(x_0, y_0)$

- $\frac{b + mx_0 - y_0}{\sqrt{1 + m^2}}$  se  $(x_0, y_0)$  si trova sotto la retta
- $\frac{-b - mx_0 + y_0}{\sqrt{1 + m^2}}$  se  $(x_0, y_0)$  si trova sopra la retta

Il programma lineare riguarda la ricerca di un cerchio che massimizzi il raggio  $r$  (distanza fra  $(x_0, y_0)$  e  $P$ ) variando fra il possibile centro  $(x_0, y_0)$  all'interno del poligono, rispettando i vincoli del tipo

- $\frac{b_l + mx_0 - y_0}{\sqrt{1 + m^2}} \geq r, \quad \forall l \in 1, \dots, k$
- $\frac{-b_l - mx_0 + y_0}{\sqrt{1 + m^2}} \geq r \quad \forall l \in k + 1, \dots, n$

## 2.3 Il Metodo del Simplexso

Il metodo del simplexso è un'algoritmo per la risoluzione dei programmi lineari. Questi devono però assumere una forma differente da quella standard introdotta nella definizione 8. Un programma lineare è definito come segue

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Per ogni generica disuguaglianza del tipo

$$a_{1i}x_1 + a_{2i}x_2 + \dots + a_{ni}x_n \leq b_i$$

Si considera una nuova variabile  $y_i \geq 0$  detta **slack**, e la disuguaglianza diventa un'uguaglianza come segue

$$a_{1i}x_1 + a_{2i}x_2 + \dots + a_{ni}x_n + y_i = b_i$$

Vi sarà quindi un vettore  $\mathbf{y} \in \mathbb{R}^m$  che viene introdotto nel problema, che assume la seguente forma

$$\begin{aligned} \max \quad & [\mathbf{c}^T \mid 0 \dots 0] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \\ \text{s.t.} \quad & [A \mid \text{Id}_m] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{b} \\ & \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \geq \mathbf{0} \end{aligned}$$

Dove  $\text{Id}_m$  è la matrice identità  $m \times m$ . Un problema LP di questo tipo è detto in **forma di equazione**, ed il metodo del simplexso opera su un programma lineare di tale forma. In generale, si scrive

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

**Assunzione** : Il programma lineare considerato ha almeno una soluzione ammissibile. Ossia  $\exists \bar{\mathbf{x}}$  tale che  $A\bar{\mathbf{x}} = \mathbf{b}$  e  $\mathbf{x} \geq \mathbf{0}$ . Data la matrice  $A$ , si considerino due righe di essa, ossia  $A_i$  e  $A_j$ , sia  $A'$  un'altra matrice identica a  $A$ , in cui la riga  $j$ -esima è sostituita con la riga  $A_i + A_j$

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{i1} & \dots & a_{in} \\ \vdots & & \vdots \\ a_{j1} & \dots & a_{jn} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \quad A' = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{i1} & \dots & a_{in} \\ \vdots & & \vdots \\ a_{j1} + a_{i1} & \dots & a_{jn} + a_{in} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \quad (2.12)$$

Se  $\mathbf{x}$  soddisfa  $A\mathbf{x} = \mathbf{b}$  allora soddisfa anche  $A'\mathbf{x} = \mathbf{b}'$  dove  $\mathbf{b}'$  è identico a  $\mathbf{b}$  ma il  $j$ -esimo elemento è sommato all' $i$ -esimo

$$\mathbf{b}' = [b_1 \dots b_i \dots b_i + b_j \dots b_n]^T$$

Le operazioni sulle righe di  $A$  non cambiano l'insieme delle soluzioni, si assume che le righe siano *linearmente indipendenti*, che il rango di  $A$  sia  $m$ , e che lo span delle righe di  $A$  è un sottospazio di  $\mathbb{R}^n$ .

**Proposizione 7** *Il rango delle colonne di  $A$  è  $m$ , la dimensione del sottospazio di  $\mathbb{R}^m$  dato dallo span delle colonne è  $m$ .*

*Dimostrazione* : **TODO**

Il problema descritto è

$$A\mathbf{x} = \mathbf{b} \quad (2.13)$$

$$\mathbf{x} \geq \mathbf{0} \quad (2.14)$$

$$\text{esiste una soluzione} \quad (2.15)$$

$$\text{rango righe di } A = \text{rango colonne di } A = m \quad (2.16)$$



Considerando  $\ker(A) = \{\mathbf{x} \text{ t.c. } A\mathbf{x} = \mathbf{0}\}$  si ha un sottospazio di dimensione  $n - m$  di  $\mathbb{R}^n$ , lo spazio delle soluzioni di  $A\mathbf{x} = \mathbf{b}$  è un **sottospazio affine** e si ottiene sommando  $\mathbf{b}$  a tutti i vettori di  $\ker(A)$ .

### 2.3.1 Soluzioni Ammissibili Basiche

*Esempio* : Si consideri il triangolo delineato dai punti  $(0, 0, 3)$ ,  $(0, 3, 0)$ ,  $(3, 0, 0)$ , tutti i punti di tale triangolo sono i vettori  $\mathbf{x}$  che soddisfano  $A\mathbf{x} = \mathbf{b}$ . Si ha  $A = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$  e  $b = 3$

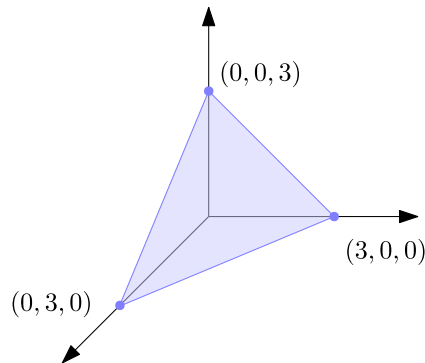


Figura 2.4: triangolo in  $\mathbb{R}^3$

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 3 \implies x + y + z = 3$$

I 3 vertici del triangolo rappresentano delle soluzioni "speciali", ottenute ponendo 2 delle 3 variabili uguali a zero e risolvendo per le altre, tali soluzioni sono dette *basiche*.

In generale,  $A \in \text{Mat}(m \times n)$  e  $\mathbf{b} \in \mathbb{R}^m$ , si vogliono porre un numero massimale di variabili pari a zero e risolvere per le altre, si pongono esattamente  $n - m$  variabili nulle.

Sia  $\mathcal{B} \subseteq \{1, \dots, n\}$  l'insieme degli indici  $i$  per cui la variabile  $x_i$  non è posta a zero, e sia  $A_{\mathcal{B}}$  la sotto-matrice di  $A$  composta dalle colonne il cui indice è relativo ad una variabile diversa da zero.

$$\begin{bmatrix} a_{1i} & \dots & a_{mi} \end{bmatrix}^T \text{ è una colonna di } A_{\mathcal{B}} \iff i \in \mathcal{B}$$

Sia  $\mathbf{b}_{\mathcal{B}}$  definito in maniera intuitiva, come il vettore contenente la componente  $j$ -esima di  $\mathbf{b}$  se  $j \in \mathcal{B}$ . Si considera il problema derivato

$$A_{\mathcal{B}}\mathbf{x}_{\mathcal{B}} = \mathbf{b}_{\mathcal{B}} \quad A_{\mathcal{B}} \in \text{Mat}(m \times m) \quad (2.17)$$

Se  $A_{\mathcal{B}}$  non è **singolare** (il determinante è diverso da zero) allora esiste una soluzione  $\bar{\mathbf{x}}$  per il problema derivato, se  $\bar{\mathbf{x}} \geq 0$ , allora è anche una soluzione per il problema originale, ponendo a zero tutte le componenti della soluzione i cui indici non sono in  $\mathcal{B}$ .

- se  $\bar{\mathbf{x}} = [x_{k1} \ x_{k2} \ \dots \ x_{km}]^T$  è soluzione di  $A_{\mathcal{B}}\mathbf{x}_{\mathcal{B}} = \mathbf{b}_{\mathcal{B}}$
- si ha  $\{k1, k2, \dots, km\} = \mathcal{B}$
- allora il vettore  $\bar{\mathbf{x}}'$  con
  - $x_i = x_{kj}$  se  $i = kj$  per qualche  $j \leq m$ , ossia  $i \in \mathcal{B}$
  - $x_i = 0$  se  $i \notin \mathcal{B}$

è soluzione di  $A\mathbf{x} = \mathbf{b}$  se  $\bar{\mathbf{x}}' \geq 0$ .

**Definizione 11** Dato un programma lineare

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Una soluzione ammissibile  $\bar{\mathbf{x}}$  è detta **basica** (BFS) se esiste un sotto-insieme di indici  $\mathcal{B} \subseteq \{1, 2, \dots, n\}$  tale per cui

- $A_{\mathcal{B}}$  non è singolare
- $\bar{x}_j = 0 \ \forall j \notin \mathcal{B}$

Esempio : Data la matrice

$$A = \begin{bmatrix} 1 & 6 & 3 & -3 & 4 \\ 0 & 1 & 3 & 2 & 4 \end{bmatrix} \quad (2.18)$$

Ed il sistema di equazioni

$$A\mathbf{x} = \begin{bmatrix} 6 \\ 6 \end{bmatrix}$$

Per la base  $\mathcal{B} = \{2, 4\}$  si ha

$$A_{\mathcal{B}} = \begin{bmatrix} 6 & -3 \\ 1 & 2 \end{bmatrix}$$

La matrice non è singolare, per il sistema

$$\begin{bmatrix} 6 & -3 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \end{bmatrix}$$

una soluzione è

$$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Quindi una soluzione ammissibile basica per il problema originale è

$$\bar{\mathbf{x}} = [0 \quad 2 \quad 0 \quad 2 \quad 0]^T$$

Se considerassimo la base  $\mathcal{B}' = \{3, 5\}$ , si avrebbe la sotto-matrice

$$A_{\mathcal{B}'} = \begin{bmatrix} 3 & 4 \\ 3 & 4 \end{bmatrix}$$

Essendo singolare, per tale base non esiste alcuna soluzione basica.

**Proposizione 8** Sia  $A\mathbf{x} = \mathbf{b}$ ,  $\mathbf{x} \geq \mathbf{0}$  un programma lineare, e sia  $\bar{\mathbf{x}}$  una BFS (soluzione ammissibile basica), si consideri l'insieme di indici

$$K = \{i \text{ t.c. } \bar{x}_i > 0\}$$

Le colonne della matrice  $A_K$  sono linearmente indipendenti.

*Dimostrazione* : La dimostrazione è banale, esiste una base  $\mathcal{B}$  a cui  $\bar{\mathbf{x}}$  fa riferimento per cui  $K \subseteq \mathcal{B}$ , essendo le colonne di  $A_{\mathcal{B}}$  linearmente indipendenti, anche un loro sotto-insieme, ossia quelle di  $A_K$  lo sono. ■

**Proposizione 9** Si consideri un programma lineare, sia  $\bar{\mathbf{x}}$  una soluzione ammissibile e

$$K = \{i \text{ t.c. } \bar{x}_i > 0\}$$

$\bar{\mathbf{x}}$  è una BFS se e solo se le colonne di  $A_K$  sono linearmente indipendenti.

*Dimostrazione* : Verrà dimostrata solo una delle due implicazioni, si assume che  $\bar{\mathbf{x}}$  è una soluzione ammissibile e che le colonne di  $A_K$  siano linearmente indipendenti. Per dimostrare che  $\bar{\mathbf{x}}$  sia una BFS è necessario trovare una base che contenga  $K$ .

Sia  $\mathcal{B}^*$  un'insieme definito come segue

1.  $K \subseteq \mathcal{B}^*$
2. Le colonne di  $\mathcal{B}^*$  sono linearmente indipendenti
3.  $\mathcal{B}^*$  è l'insieme massimale che rispetti le due condizioni precedenti

Bisogna mostrare che  $\mathcal{B}^*$  ha  $m$  elementi in modo tale che  $A_{\mathcal{B}^*}$  sia una matrice quadrata  $m \times m$ .

Si assuma che  $|\mathcal{B}^*| < m$ , ciò significherebbe che le restanti colonne di  $A$  sono contenute nello span delle colonne di  $A_{\mathcal{B}^*}$ , quindi  $A$  ha meno di  $m$  colonne linearmente indipendenti, ma è noto che il rango di  $A$  sia  $m$ , quindi l'assunzione  $|\mathcal{B}^*| < m$  porta ad una contraddizione  $\implies |\mathcal{B}^*| = m$ . ■

**Osservazione 7** Ogni base ha almeno una BFS associata.

**Conclusione** : Esiste un numero finito di BFS. Ogni base è un sotto-insieme di  $m$  elementi preso da  $\{1, 2, \dots, n\}$ , quindi esistono al più  $\binom{n}{m}$  basi distinte, quindi esistono al più  $\binom{n}{m}$  BFS.

Il seguente risultato è fondamentale in quanto stabilisce che un programma lineare (che riguarda la ricerca di una soluzione fra un'insieme non numerabile) si può ridurre ad una ricerca fra un'insieme finito di possibili soluzioni ottimali.

**Teorema 2** Si consideri un programma lineare in forma di equazione

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ \text{A} \mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

1. Se esiste almeno una soluzione ammissibile ed il problema è limitato (l'insieme delle soluzioni ammissibili è compatto), allora esiste una soluzione ottimale.
2. Se esiste una soluzione ottimale, allora esiste anche una BFS che è a sua volta ottimale.

La dimostrazione del teorema 2 richiede alcuni passi preliminari.

**Statement  $\star$**  : Se la funzione obiettivo di un LP in forma di equazione è limitata superiormente, allora per ogni soluzione ammissibile  $\mathbf{y}$  esiste una BFS  $\mathbf{z}$  tale che  $\mathbf{c}^T \mathbf{z} \geq \mathbf{c}^T \mathbf{y}$ .

**Claim 1** : Se  $\star$  è vera, allora il teorema è dimostrato.

*Dimostrazione Claim 1* : Si consideri la BFS  $\mathbf{y}$  che massimizza la funzione obiettivo rispetto tutte le altre BFS. Assumiamo che  $\mathbf{y}$  non sia ottimale, allora esiste  $\mathbf{y}^*$  (che non è una BFS) tale che

$$\mathbf{c}^T \mathbf{y}^* \geq \mathbf{c}^T \mathbf{y}$$

Ma per il claim  $\star$  esiste una BFS  $\mathbf{z}$  tale per cui

$$\mathbf{c}^T \mathbf{z} \geq \mathbf{c}^T \mathbf{y}^* \geq \mathbf{c}^T \mathbf{y}$$

Ciò va in contraddizione con l'assunzione che  $\mathbf{y}$  sia massimale fra le BFS, quindi non può esistere tale  $\mathbf{y}^* \implies \mathbf{y}$  è una BFS ottimale  $\implies$  il claim è dimostrato :  $\star \implies$  teorema 2. ■

Per dimostrare il teorema è quindi sufficiente dimostrare lo Statement  $\star$ . Sia  $\mathbf{y}$  un'arbitraria soluzione ammissibile, sia  $\mathbf{z}$  un'altra soluzione ammissibile tale che

1.  $\mathbf{c}^T \mathbf{z} \geq \mathbf{c}^T \mathbf{y}$
2.  $\mathbf{z}$  ha un numero massimale di componenti uguali a zero

Tale  $\mathbf{z}$  esiste dato che nel caso peggiore si può considerare  $\mathbf{z} = \mathbf{y}$ . Si consideri l'insieme  $K$  degli indici delle componenti positive di  $\mathbf{z}$

$$K = \{j \text{ t.c. } z_j > 0\}$$

La proposizione 8 afferma che se le colonne di  $A_K$  sono linearmente indipendenti  $\mathbf{z}$  è una BFS, ciò dimostrerebbe lo Statement  $\star$ .

Si assuma che le colonne di  $A_K$  siano linearmente dipendenti. Sia  $k = |K|$ . Esistono dei coefficienti  $\alpha_1, \dots, \alpha_k$  tali per cui

$$\alpha_1 \begin{bmatrix} A_K^1 \end{bmatrix} + \dots + \alpha_k \begin{bmatrix} A_K^k \end{bmatrix} = \mathbf{0} \implies \quad (2.19)$$

$$A_K \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \mathbf{0} \quad (2.20)$$

Si consideri un particolare vettore  $\mathbf{w}$  definito come segue



- la componente  $j$ -esima di  $\mathbf{w}$  contiene  $\alpha_j$  se  $j \in K$ ,  $\alpha_j$  è il coefficiente moltiplicato alla colonna  $j$ -esima di  $A_K$  nella combinazione lineare dell'equazione 2.19.
- Se  $j \notin K$ , allora la componente  $j$ -esima di  $\mathbf{w}$  è 0.

Data l'equazione 2.20, risulta chiaro che il prodotto della matrice originale  $A$  e  $\mathbf{w}$  sia uguale al prodotto fra  $A_K$  ed il vettore dei coefficienti  $\alpha_1 \dots, \alpha_k$ .

$$A\mathbf{w} = A_K \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \mathbf{0} \quad (2.21)$$

Riconsiderando la soluzione  $\mathbf{z}$ , si può sommare a  $\mathbf{w}$  e moltiplicare per la matrice  $A$

$$A(\mathbf{z} + \mathbf{w}) = A\mathbf{z} + A\mathbf{w} = A\mathbf{z} + \mathbf{0} = \mathbf{b} + \mathbf{0} = \mathbf{b}$$

Essendo che  $\mathbf{z}$  è soluzione il prodotto fra quest'ultimo ed  $A$  è uguale al vettore  $\mathbf{b}$  relativo ai vincoli del programma lineare. Nonostante  $\mathbf{z} + \mathbf{w}$  soddisfi i vincoli, non è certo che sia soluzione, in quanto non è certo se abbia o no componenti negative (si ricordi che una soluzione deve essere maggiore o uguale al vettore  $\mathbf{0}$ ).

Si noti come per definizione di  $\mathbf{w}$ , questo ha una componente uguale a zero in una data posizione, se in quella data posizione la soluzione  $\mathbf{z}$  ha la componente uguale a zero, quindi  $\mathbf{w}$  ha *almeno tanti* zeri quanti quelli di  $\mathbf{z}$ .

**Claim 2 :** Il vettore  $\mathbf{w}$  soddisfa le seguenti

1.  $\mathbf{c}^T \mathbf{w} \geq 0$
2.  $\exists j \in K$  t.c.  $w_j < 0$

*Dimostrazione :* Se per  $\mathbf{w}$  il punto (2) non dovesse essere soddisfatto, si potrebbe considerare  $-\mathbf{w}$ , questo soddisfa tutte le condizioni per cui è stato definito  $\mathbf{w}$  :  $A(-\mathbf{w}) = A\mathbf{w} = \mathbf{0}$ . Tale sostituzione si può considerare anche per soddisfare il punto (1), nel caso in cui si dovesse verificare che  $\mathbf{c}^T \mathbf{w} < 0$ .

Se la condizione (2) non dovesse essere soddisfatta, allora si consideri il vettore (in funzione di  $t \in \mathbb{R}^+$ )

$$\mathbf{z}(t) = \mathbf{z} + t\mathbf{w}$$

Tale vettore soddisfa i vincoli del programma lineare

$$A\mathbf{z}(t) = A(\mathbf{z} + t\mathbf{w}) = A\mathbf{z} + tA\mathbf{w} = \mathbf{b} + t \cdot \mathbf{0} = \mathbf{b}$$

Inoltre essendo la (2) non soddisfatta, si ha che, per ogni possibile  $t \in \mathbb{R}^+$

$$z(t)_j = z_j + tw_j \geq 0$$

questo perché

1.  $z_j \geq 0$  essendo  $\mathbf{z}$  una soluzione
2.  $t$  è non negativo perché varia in  $\mathbb{R}^+$
3.  $w_j$  non è negativo dato che la (2) non è soddisfatta

Essendo  $\mathbf{z}(t)$  sempre maggiore o uguale a zero si ha:

$$\mathbf{c}^T \cdot \mathbf{z}(t) = \mathbf{c}^T (\mathbf{z} + t\mathbf{w}) = \mathbf{c}^T \mathbf{z} + t\mathbf{c}^T \mathbf{w} \quad (2.22)$$

Attenzione, essendo il punto (1) soddisfatto,  $\mathbf{c}^T \mathbf{w}$  è positivo, quindi dato il termine  $t\mathbf{c}^T \mathbf{w}$ , la funzione  $\mathbf{c}^T \cdot \mathbf{z}(t)$  non è limitata, ciò è una contraddizione, quindi le due assunzioni del claim sono vere. ■

Dato ciò, esiste un indice  $j \in K$  per cui  $w_j < 0$ , si consideri nuovamente il vettore

$$\mathbf{z}(t) = \mathbf{z} + t\mathbf{w}$$

Avendo verificato che questo soddisfa i vincoli  $\forall t$ , ci si pone la domanda, che valore assume nella funzione obiettivo?

$$\mathbf{c}^T \mathbf{z}(t) = \mathbf{c}^T \mathbf{z} + t\mathbf{c}^T \mathbf{w}$$

Sappiamo che

- $\mathbf{c}^T \mathbf{z} \geq \mathbf{c}^T \mathbf{y}$  per la scelta di  $\mathbf{z}$
- $t\mathbf{c}^T \mathbf{w} \geq 0$  per il claim

Ne consegue che

$$\mathbf{c}^T \mathbf{z}(t) \geq \mathbf{c}^T \mathbf{y}$$

Per la chiarezza della dimostrazione, si ricordi che  $\mathbf{z}$  è stato scelto per soddisfare le seguenti

1.  $\mathbf{c}^T \mathbf{z} \geq \mathbf{c}^T \mathbf{y}$
2.  $\mathbf{z}$  ha un numero massimale di componenti uguali a zero

$$\mathbf{z}(t) = \begin{bmatrix} 0 \\ \vdots \\ z_i \\ \vdots \\ 0 \end{bmatrix} + t \begin{bmatrix} 0 \\ \vdots \\ w_i \\ \vdots \\ 0 \end{bmatrix}$$

sia  $w_i$  una componente negativa di  $\mathbf{w}$ , se  $t = -\frac{z_i}{w_i}$  allora

$$\mathbf{z}\left(-\frac{z_i}{w_i}\right) = \begin{bmatrix} 0 \\ \vdots \\ z_i \\ \vdots \\ 0 \end{bmatrix} - \frac{z_i}{w_i} \begin{bmatrix} 0 \\ \vdots \\ w_i \\ \vdots \\ 0 \end{bmatrix} \implies z\left(-\frac{z_i}{w_i}\right)_i = z_i - z_i = 0$$

Il vettore  $\mathbf{z}\left(-\frac{z_i}{w_i}\right)$  ha una componente nulla in più rispetto a  $\mathbf{z}$ , se si considera

$$t^* = \min_{i \text{ t.c. } w_i < 0} \left(-\frac{z_i}{w_i}\right)$$

il vettore  $\mathbf{z}(t^*)$  risulterà non negativo, quindi:

$$\begin{aligned} A\mathbf{z}(t^*) &= \mathbf{b} \\ \mathbf{z}(t^*) &\geq 0 \end{aligned}$$

Il vettore  $\mathbf{z}(t^*)$  è una soluzione ammissibile ed ha uno 0 in più rispetto a  $\mathbf{z}$ , ma questa è una contraddizione (data la scelta di  $\mathbf{z}$ ), quindi è impossibile che le colonne di  $A_K$  siano linearmente dipendenti  $\implies \mathbf{z}$  è una BFS, lo Statement  $\star$  è dimostrato. ■

**Corollario :** Se un programma lineare ha una funzione obiettivo limitata, allora esiste un'algoritmo che trova una soluzione ottimale in un numero finito di passi.

### 2.3.2 Ricerca sul Poliedro

Si consideri il seguente programma lineare

$$\begin{aligned} \max \quad & x_1 + x_2 \\ -x_1 + x_2 & \leq 1 \\ x_1 & \leq 3 \\ x_2 & \leq 2 \\ x_1, x_2 & \geq 0 \end{aligned}$$

L'insieme delle soluzioni ammissibili è mostrato in figura 2.3.2. Per trasformare il problema in forma di equazione occorre aggiungere 3 variabili slack  $x_3, x_4, x_5$ .

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ -x_1 + x_2 + x_3 & = 1 \\ x_1 + x_4 & = 3 \\ x_2 + x_5 & = 2 \\ x_1, x_2, x_3, x_4, x_5 & \geq 0 \end{aligned}$$

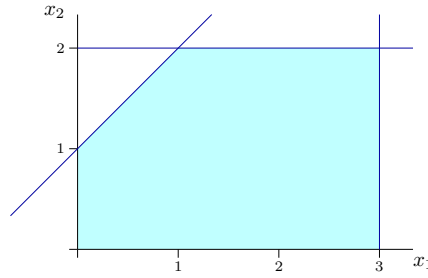


Figura 2.5: Insieme definito dai vincoli

in forma matriciale:

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{c} = [1 \quad 1 \quad 0 \quad 0 \quad 0]^T$$

Si consideri la base  $\mathcal{B} = \{1, 2, 4\}$  :

$$A_{\mathcal{B}} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

è non singolare, si può risolvere il sistema di equazioni

$$\begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} \implies \begin{cases} x_1 = 1 \\ x_2 = 2 \\ x_4 = 2 \end{cases} \quad (2.23)$$

La BFS è  $\mathbf{x} = [1 \quad 2 \quad 0 \quad 2 \quad 0]^T$ . Le variabili slack  $x_3, x_5$  della soluzione  $\mathbf{x}$  sono poste a zero, ciò significa che tale punto soddisfa i vincoli del problema originale (in forma di uguaglianza), in particolare, soddisfa le uguaglianze del primo e del terzo vincolo

$$\begin{aligned} -x_1 + x_2 &= 1 \\ x_2 &= 2 \end{aligned}$$

Tale punto può essere geometricamente individuato nell'insieme delle soluzioni ammissibili del problema originale, ed equivale al punto su cui si intersecano le due rette che definiscono il primo ed il terzo vincolo, come mostrato in figura 2.3.2.

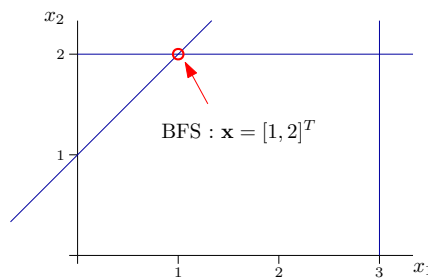


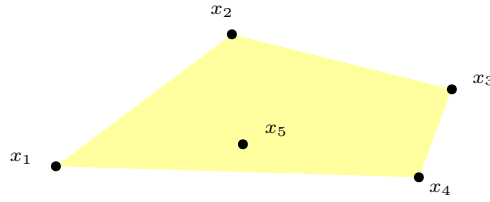
Figura 2.6: punto di intersezione

**Definizione 12** Un'insieme  $\mathcal{X} \subset \mathbb{R}^n$  è **aperto** se  $\forall \mathbf{x} \in \mathcal{X}, \exists \epsilon > 0$  tale che  $\{\mathbf{y} \in \mathbb{R}^n \mid \|\mathbf{y} - \mathbf{x}\| \leq \epsilon\} \subset \mathcal{X}$ .

**Definizione 13** Un'insieme  $\mathcal{X} \subset \mathbb{R}^n$  è **chiuso** se il suo complemento  $\mathbb{R}^n \setminus \mathcal{X}$  è un'insieme aperto.

**Osservazione 8** L'unione di due insiemi aperti è un'insieme aperto. L'intersezione di due insiemi chiusi è un'insieme chiuso.

**Definizione 14** Dato un sotto-insieme di punti  $I \subset \mathbb{R}^n$ , si definisce il suo **inviluppo convesso** l'intersezione di tutti i sotto-insiemi convessi di  $\mathbb{R}^n$  contenenti  $I$ . Alternativamente, si può dire che l'inviluppo convesso di  $I$  è il più piccolo insieme convesso contenente  $I$ .

Figura 2.7: inviluppo convesso dei punti  $x_1, x_2, \dots, x_5$ 

**Definizione 15** Dati  $n$  punti  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^m$ , una loro **combinazione convessa** è un punto  $\mathbf{z}$  definito come segue

- $\mathbf{z} = \sum_{i=1}^n \alpha_i \mathbf{x}_i$
- $\alpha_i \geq 0 \quad \forall i$
- $\sum_{i=1}^n \alpha_i = 1$

**Osservazione 9** Una combinazione convessa fra due punti è un segmento di linea.

**Proposizione 10** Siano  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  dei punti in  $\mathbb{R}^m$ , sia  $C$  l'inviluppo convesso di tali punti, e sia  $\tilde{C}$  l'insieme di tutte le combinazioni convesse

$$\tilde{C} = \left\{ \sum_{i=1}^n \alpha_i \mathbf{x}_i \text{ t.c. } \alpha_i \geq 0 \quad \forall i, \quad \sum_{i=1}^n \alpha_i = 1 \right\}$$

si ha che  $C = \tilde{C}$

*Dimostrazione* : La dimostrazione procederà classicamente con una doppia inclusione. Si vuole mostrare come prima cosa che  $C \subseteq \tilde{C}$ , essendo  $C$  l'intersezione di tutti gli insiemi convessi contenenti i punti, ed essendo che  $\tilde{C}$  contiene ogni punto  $\mathbf{x}_i$ , è sufficiente mostrare che  $\tilde{C}$  sia convesso.

Siano  $\mathbf{z}_1, \mathbf{z}_2 \in \tilde{C}$ , ossia della forma

$$\begin{aligned} \mathbf{z}_1 &= \sum_i \alpha_i \mathbf{x}_i, & \alpha_i &\geq 0, \quad \sum_i \alpha_i = 1 \\ \mathbf{z}_2 &= \sum_i \beta_i \mathbf{x}_i, & \beta_i &\geq 0, \quad \sum_i \beta_i = 1 \end{aligned}$$

Un generico punto sul segmento di  $\mathbf{z}_1, \mathbf{z}_2$  è

$$t\mathbf{z}_1 + (1-t)\mathbf{z}_2, \quad t \in [0, 1]$$

esplicitando

$$t \sum_i \alpha_i \mathbf{x}_i + (1-t) \sum_i \beta_i \mathbf{x}_i = \tag{2.24}$$

$$\sum_i t\alpha_i \mathbf{x}_i + \sum_i (1-t)\beta_i \mathbf{x}_i = \tag{2.25}$$

$$\sum_i (t\alpha_i + (1-t)\beta_i) \mathbf{x}_i \tag{2.26}$$

Bisogna mostrare che  $\sum_i (t\alpha_i + (1-t)\beta_i) \mathbf{x}_i$  è una combinazione convessa, essendo che

$$t \geq 0, \quad \alpha_i \geq 0, \quad \beta_i \geq 0$$

è immediato che, per ogni  $i$  si ha che  $t\alpha_i + (1-t)\beta_i \geq 0$ , inoltre

$$\sum_i t\alpha_i + (1-t)\beta_i = \tag{2.27}$$

$$\sum_i t\alpha_i + \sum_i (1-t)\beta_i = \tag{2.28}$$

$$t \sum_i \alpha_i + (1-t) \sum_i \beta_i \tag{2.29}$$

Essendo che per ipotesi  $\sum_i \alpha_i = \sum_i \beta_i = 1$ , si ha che

$$t \sum_i \alpha_i + (1-t) \sum_i \beta_i = t + (1-t) = 1 \quad (2.30)$$

Quindi ogni punto sul segmento  $t\mathbf{z}_1 + (1-t)\mathbf{z}_2$ ,  $t \in [0, 1]$  è a sua volta una combinazione convessa di  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \implies \tilde{C}$  è convesso  $\implies C \subseteq \tilde{C}$ .

Si vuole mostrare ora che  $\tilde{C} \subseteq C$ , sia  $\mathbf{z}$  una combinazione convessa,  $\mathbf{z} = \sum_i \alpha_i \mathbf{x}_i$ , si procede per induzione sul numero di coefficienti  $\alpha_i$  il cui valore è diverso da zero.

- *Caso base* : Solamente un coefficiente  $\alpha_i$  è diverso da zero, sia questo  $\alpha_j$  ( $j$  fissato), allora

$$\sum_i \alpha_i \mathbf{x}_i = 0 \cdot \mathbf{x}_0 + \dots 1 \cdot \mathbf{x}_j + \dots 0 \cdot \mathbf{x}_n = \mathbf{x}_j$$

Essendo che i punti sono contenuti nell'involuppo convesso, si ha che  $\mathbf{z} \in C$

- *Secondo caso base* : Se il numero di coefficienti di  $\mathbf{z}$  diversi da zero fosse 2, allora la combinazione convessa sarebbe del tipo

$$\mathbf{z} = t\mathbf{x}_i + (1-t)\mathbf{x}_j, \quad \text{per qualche } i, j$$

anche in questo caso  $\mathbf{z}$  si troverebbe in  $C$ , dato che  $C$  è convesso e  $\mathbf{x}_i, \mathbf{x}_j \in C$ , come mostrato nel caso base.

- *Ipotesi induttiva* : Le combinazioni convesse con  $k-1$  coefficienti diversi da zero sono in  $C$ .
- *Passo induttivo* : Sia  $\mathbf{z}$  una combinazione convessa con  $k$  coefficienti diversi da zero. È sufficiente mostrare che  $\mathbf{z}$  si trova sul segmento di linea di due punti contenuti in  $C$ . Sia  $j$  un fissato indice tale per cui il coefficiente  $\alpha_j$  di  $\mathbf{z}$  è diverso da zero, per definizione si ha che

$$\sum_{i=1, i \neq j}^n \alpha_i = 1 - \alpha_j$$

Questo comporta che

$$\frac{1}{1 - \alpha_j} \sum_{i=1, i \neq j}^n \alpha_i = 1$$

Si considerino  $n$  coefficienti  $\alpha'_1, \dots, \alpha'_n$  definiti come segue

$$\alpha'_i = \begin{cases} 0 & \text{se } i = j \\ \frac{\alpha_i}{1 - \alpha_j} & \text{altrimenti} \end{cases}$$

Si consideri la combinazione convessa  $\mathbf{z}'$  data da tali coefficienti

$$\mathbf{z}' = \sum_{i=1}^n \alpha'_i \mathbf{x}_i$$

Chiaramente,  $\mathbf{z}'$  ha  $k-1$  coefficienti diversi da zero, quindi per ipotesi induttiva  $\mathbf{z}' \in C$ . Si consideri il seguente punto sul segmento fra  $\mathbf{z}'$  e  $\mathbf{x}_j$  :

$$(1 - \alpha_j)\mathbf{z}' + \alpha_j \mathbf{x}_j$$

Essendo sul segmento, tale punto è contenuto in  $C$ , esplicitando:

$$(1 - \alpha_j)\mathbf{z}' + \alpha_j \mathbf{x}_j = \quad (2.31)$$

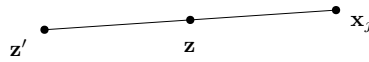
$$(1 - \alpha_j) \sum_{i=1}^n \alpha'_i \mathbf{x}_i + \alpha_j \mathbf{x}_j = \quad (2.32)$$

$$\sum_{i=1}^n (1 - \alpha_j) \alpha'_i \mathbf{x}_i + \alpha_j \mathbf{x}_j = \quad (2.33)$$

$$\sum_{i=1, i \neq j}^n \alpha_i \mathbf{x}_i + \alpha_j \mathbf{x}_j = \mathbf{z} \quad (2.34)$$

Ma allora  $\mathbf{z}$  si trova sul segmento di linea fra  $\mathbf{z}'$  e  $\mathbf{x}_j$ , quindi  $\mathbf{z} \in C \implies \tilde{C} \subseteq C$ , questo completa la dimostrazione,  $C = \tilde{C}$ . ■



Figura 2.8: locazione geometrica di  $\mathbf{z}$ 

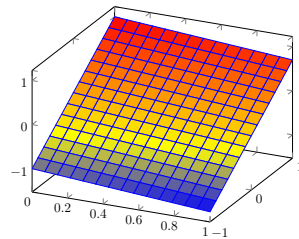
**Definizione 16** Un **iperpiano** in  $\mathbb{R}^n$  è un sottospazio affine di dimensione  $n - 1$  definito dall'insieme dei punti che soddisfano un'equazione del tipo

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$$

Ogni iperpiano definisce due *mezzi spazi*, ossia due insiemi convessi e chiusi, la cui unione comprende tutto  $\mathbb{R}^n$ , definiti dai punti che soddisfano le equazioni

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$$

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b$$

Figura 2.9: iperpiano in  $\mathbb{R}^3$ 

**Definizione 17** Un **poliedro** è l'intersezione di un numero finito di mezzi-spazi definiti da iperpiani. La dimensione del poliedro  $P$  è uguale alla dimensione del più piccolo sotto-spazio affine di  $\mathbb{R}^n$  contenente  $P$ .

**Definizione 18** Un **politopo**  $P$  è un poliedro limitato, ossia,  $\exists c \in \mathbb{R}$  tale che  $\forall \mathbf{x} \in P, \|\mathbf{x}\| \leq c$ .

**Definizione 19** Dato un poliedro  $P$ , un punto  $\mathbf{v}$  è un **vertice** di  $P$  se esiste un iperpiano

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \text{ t.c. } a_1x_1 + a_2x_2 + \cdots + a_nx_n = b\}$$

tale per cui

- $\mathbf{v} \in \mathcal{X}$
- $\forall \bar{\mathbf{x}} \in P, \bar{\mathbf{x}} = [\bar{x}_1, \dots, \bar{x}_n]^T \neq \mathbf{v}$  si ha

$$a_1\bar{x}_1 + a_2\bar{x}_2 + \cdots + a_n\bar{x}_n < b$$

**Definizione 20** Dato un poliedro  $P$ , una **faccia** è un'insieme  $\mathcal{U} \subseteq P$  tale per cui esiste un iperpiano  $\mathcal{X}$  tale che  $\mathcal{X} \cap P = \mathcal{U}$ , e per cui, ogni punto di  $P$  si trova in uno dei due mezzi spazi definiti da  $\mathcal{X}$

La dimensione di una faccia è uguale alla dimensione del più piccolo sottospazio affine che la contiene.

**Definizione 21** Un **angolo** è una faccia di dimensione 1.

**Teorema 3** Se  $P$  è il poliedro rappresentante l'insieme delle soluzioni ammissibili di un programma lineare in forma di equazione, allora  $\mathbf{v}$  è un vertice di  $P$  se e solo se  $\mathbf{v}$  è una soluzione ammissibile basica per il programma lineare.

*Dimostrazione* : Verranno dimostrati entrambe le implicazione del *se e solo se*.

$\Rightarrow$  : Sia vuole dimostrare che il vertice di un poliedro è soluzione di un programma lineare. Sia  $\mathbf{v}$  un vertice di  $P$ , per definizione, esiste un iperpiano  $a_1x_1 + \dots + a_nx_n = b$  tale per cui  $P$  è contenuto in una delle due metà definite da esso, sia questa  $a_1x_1 + \dots + a_nx_n \leq b$ , inoltre

$$a_1v_1 + \dots + a_nv_n = b$$

e

$$a_1x_1 + \dots + a_nx_n < b, \quad \forall \mathbf{x} \in P, \quad \mathbf{x} \neq \mathbf{v}$$

Quindi, considerando il programma lineare

$$\max [a_1 \quad a_2 \quad \dots \quad a_n] \mathbf{x} < b \quad (2.35)$$

$$\mathbf{x} \in P \quad (2.36)$$

$\mathbf{v}$  è l'unica soluzione ottima, per il teorema 2, allora  $\mathbf{v}$  è una BFS.

$\Leftarrow$  : Sia  $\mathbf{v}$  una BFS per un dato programma lineare, il cui insieme delle soluzioni ammissibili è il poliedro  $P$ , a  $\mathbf{v}$  è associata una base  $\mathcal{B} \subset \{1, 2, \dots, n\}$ , si consideri vettore  $\tilde{\mathbf{c}}$  definito come segue

$$\tilde{\mathbf{c}} = \begin{bmatrix} \tilde{c}_1 \\ \vdots \\ \tilde{c}_n \end{bmatrix} \quad \tilde{c}_i = \begin{cases} 0 & \text{se } i \in \mathcal{B} \\ -1 & \text{se } i \notin \mathcal{B} \end{cases} \quad (2.37)$$

Essendo che ogni componente  $i$ -esima di  $\mathbf{v}$  è nulla se  $i \notin \mathcal{B}$ , è immediato che

$$\tilde{\mathbf{c}}^T \mathbf{v} = 0$$

Inoltre, preso un qualsiasi altro punto  $\mathbf{x} \in P$ , se  $\exists j$  tale che  $x_j > 0$ , allora  $\tilde{\mathbf{c}}^T \mathbf{x} < 0$ .

**Osservazione 10**  $\forall \mathbf{x} \in P, \quad \tilde{\mathbf{c}}^T \mathbf{x} \leq 0$

Quindi, considerando l'iperpiano

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \text{ t.c. } \tilde{\mathbf{c}}^T \mathbf{x} = 0\}$$

si noti come

- Per l'osservazione 10 tutti i punti del poliedro  $P$  sono contenuti in una delle metà definite dall'iperpiano  $\mathcal{X}$
- $\tilde{\mathbf{c}}^T \mathbf{v} = 0$

Se  $\mathbf{v}$  fosse l'unico punto per cui  $\tilde{\mathbf{c}}^T \mathbf{v} = 0$  allora sarebbe per definizione un vertice di  $P$ . Si assuma che esiste un  $\mathbf{y} \in P$  tale per cui  $\tilde{\mathbf{c}}^T \mathbf{y} = 0$ , ciò significa che  $\forall j \notin \mathcal{B}, y_j = 0$ , questo significa che, considerando la matrice  $A_{\mathcal{B}}$ , ed il sistema di equazioni

$$A_{\mathcal{B}} \mathbf{x}_{\mathcal{B}} = \mathbf{b}_{\mathcal{B}}$$

il vettore  $\mathbf{y}_{\mathcal{B}}$  (sotto-vettore di  $\mathbf{y}$  le cui componenti sono quelle di indice contenuto in  $\mathcal{B}$ ) è soluzione, ma anche  $\mathbf{v}_{\mathcal{B}}$  è soluzione, essendo  $A_{\mathcal{B}}$  una matrice quadrata non singolare, il sistema ammette un'unica soluzione, quindi  $\mathbf{v}_{\mathcal{B}} = \mathbf{y}_{\mathcal{B}} \implies \mathbf{v} = \mathbf{y}$ , ma allora  $\mathbf{v}$  è l'unica soluzione per cui  $\tilde{\mathbf{c}}^T \mathbf{v} = 0$ , ne consegue che è un vertice. ■

### 2.3.3 La Procedura di Risoluzione

Le proposizioni ed i teoremi presentati nei paragrafi precedenti dovrebbero aver fornito un'idea di come un programma lineare si riduce alla ricerca delle soluzioni ottimali fra i vertici del poliedro definito dall'insieme delle soluzioni ammissibili.

Si consideri il programma lineare definito all'inizio della sezione 2.3.2

$$\begin{aligned} \max \quad & x_1 + x_2 \\ \text{s.t.} \quad & -x_1 + x_2 \leq 1 \\ & x_1 \leq 3 \\ & x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Il poliedro in questione è mostrato in figura 2.3.2. La soluzione ottimale si trova sul vertice in alto a destra, ossia  $\mathbf{x} = [3, 2]^T$ , presenteremo la procedura del metodo del simplesso su tale programma lineare.

Prima di procedere è necessario aggiungere delle variabili slack e trasformare il problema in forma di equazione:

$$\begin{aligned} \max \mathbf{c}^T \mathbf{x} \\ -x_1 + x_2 + x_3 &= 1 \\ x_1 + x_4 &= 3 \\ x_2 + x_5 &= 2 \\ x_1, x_2, x_3, x_4, x_5 &\geq 0 \end{aligned}$$

in forma matriciale:

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{c} = [1 \quad 1 \quad 0 \quad 0 \quad 0]^T$$

**Passo 1 :** Si parte sempre da una possibile base, sia questa  $\mathcal{B}_1 = \{3, 4, 5\}$ , si riscrive il problema risolvendo il sistema di equazioni definito da  $A$  per le variabili di base

$$\begin{aligned} x_3 &= 1 + x_1 - x_2 \\ x_4 &= 3 - x_1 \\ x_5 &= 2 - x_2 \end{aligned}$$

Il valore della funzione obiettivo è  $\mathbf{c}^T \mathbf{x} = z = x_1 + x_2$ , si scrive tale equazione insieme alle equazioni del sistema, costruendo una tabella (la cui definizione formale verrà fornita in seguito):

$x_3 = 1 + x_1 - x_2$
$x_4 = 3 - x_1$
$x_5 = 2 - x_2$
$z = x_1 + x_2$

Essendo  $x_1, x_2$  due variabili non di base, queste sono poste uguali a zero, il valore della funzione obiettivo è quindi 0, e la BFS associata a tale base si ottiene risolvendo il sistema di equazioni

$$\begin{cases} x_3 = 1 + 0 - 0 \\ x_4 = 3 - 0 \\ x_5 = 2 - 0 \end{cases} \implies \begin{cases} x_3 = 1 \\ x_4 = 3 \\ x_5 = 2 \end{cases}$$

$$\mathcal{B}_1 = \{3, 4, 5\} \implies \text{BFS} = [0 \quad 0 \quad 1 \quad 3 \quad 2]^T \implies z = 0$$

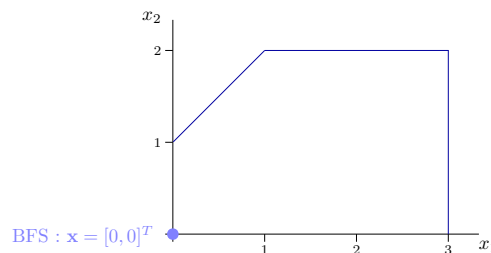


Figura 2.10: BFS con  $\mathcal{B}_1$

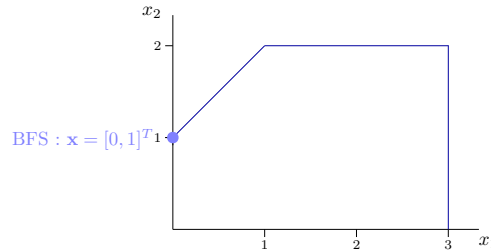
**Passo 2 :** Si considera adesso una nuova base, partendo da quella iniziale, si tira fuori una variabile per inserirne un'altra, in questo caso lo scambio avviene fra  $x_2$  e  $x_3$ , considerando  $\mathcal{B}_2 = \{2, 4, 5\}$ , la tabella diviene

$x_2 = 1 + x_1 - x_3$
$x_4 = 3 - x_1$
$x_5 = 2 - x_2$
$z = x_1 + 1 + x_1 - x_3$

Essendo  $x_1 = x_3 = 0$ , la funzione obiettivo assume valore  $z = 1$ , la BFS associata si ottiene risolvendo il sistema di equazioni

$$\begin{cases} x_2 = 1 + 0 - 0 \\ x_4 = 3 - 0 \\ x_5 = 2 - x_2 \end{cases} \implies \begin{cases} x_2 = 1 \\ x_4 = 3 \\ x_5 = 2 - 1 \end{cases} \implies \begin{cases} x_2 = 1 \\ x_4 = 3 \\ x_5 = 1 \end{cases}$$

$$\mathcal{B}_2 = \{2, 4, 5\} \implies \text{BFS} = [0 \ 1 \ 0 \ 3 \ 1]^T \implies z = 1$$

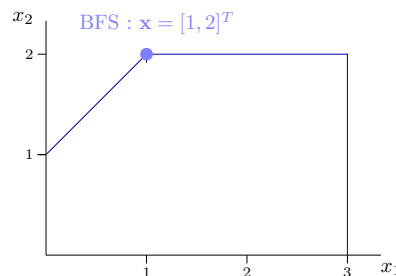
Figura 2.11: BFS con  $\mathcal{B}_2$ 

**Passo 3 :** Si vuole incrementare il valore della funzione obiettivo, si scambia la variabile  $x_1$  con la variabile  $x_5$ , ottenendo la base  $\mathcal{B}_3 = \{1, 2, 4\}$ , la tabella considerata è

$x_1 = 1 + x_3 - x_5$
$x_2 = 2 - x_5$
$x_4 = 2 - x_3 + x_5$
$z = 3 + x_3 - 2x_5$

Risolvendo il sistema di equazioni e sostituendo  $x_3, x_5$  con 0, si ha

$$\mathcal{B}_3 = \{1, 2, 4\} \implies \text{BFS} = [1 \ 2 \ 0 \ 2 \ 0]^T \implies z = 3$$

Figura 2.12: BFS con  $\mathcal{B}_3$ 

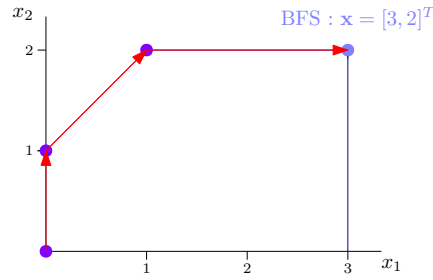
**Passo 4 :** Il seguente passo è l'ultimo, si sostituisce  $x_4$  con  $x_3$ , considerando la base  $\mathcal{B}_4 = \{1, 2, 3\}$ , la tabella risulta essere

$x_1 = 3 - x_4$
$x_2 = 2 - x_5$
$x_3 = x_5 - x_4 + 2$
$z = 5 - x_4 - x_5$

Questa risulta essere la soluzione ottimale,  $z$  non può essere incrementato in nessun modo dato che è uguale a  $5 - x_4 - x_5$ , e  $x_4, x_5$  variano in  $\mathbb{R}^+$ , quindi  $z \leq 5$ , la funzione obiettivo è massimizzata e la soluzione ottimale si ottiene risolvendo il sistema di equazioni

$$\mathcal{B}_4 = \{1, 2, 3\} \implies \text{BFS} = [3 \ 2 \ 2 \ 0 \ 0]^T \implies z = 5$$

Si osservi come geometricamente, il "cammino" fra le diverse BFS date dalle basi considerate equivale ad un cammino sui vertici del poliedro. Durante l'esecuzione dell'algoritmo, ad ogni "passo", si deve rimuovere una variabile dalla base ed inserirne un'altra, si definisce *regola del pivot* la regola con la quale si selezionano le variabili da scambiare (verranno formalizzate in seguito).

Figura 2.13: BFS con  $\mathcal{B}_4$  (soluzione ottimale)

## 2.4 Il Tableau del Simplexso

**Definizione 22** Dato un LP in forma d'equazione

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Dove  $A \in \text{Mat}(m \times n)$ , ed una base ammissibile  $\mathcal{B}$ , si definisce **tableau del simplexso** un sistema di  $m+1$  equazioni nelle variabili  $\mathbf{x}$  che ha lo stesso insieme di soluzioni di

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ z &= \mathbf{c}^T \mathbf{x} \end{aligned}$$

In forma matriciale si rappresenta come segue

$\mathbf{x}_{\mathcal{B}} = \mathbf{p} + Q\mathbf{x}_N$
$z = z_0 + \mathbf{r}^T \mathbf{x}_N$

In questo contesto  $N = \{1, 2, \dots, n\} \setminus \mathcal{B}$ , ed  $\mathbf{x}_N$  è il vettore  $\mathbf{x}$  comprendente solamente le componenti di indice  $i \in N$ .  $\mathbf{p}$  è un vettore in  $\mathbb{R}^m$ ,  $\mathbf{r}$  è un vettore in  $\mathbb{R}^{n-m}$ ,  $z_0$  è un numero reale e  $Q$  è una matrice  $m \times (n-m)$ . Data la base  $\mathcal{B}$ , si denota  $\mathcal{T}(\mathcal{B})$  il tableau ad essa relativo. Si può trovare la BFS associata a  $\mathcal{B}$  ponendo

$$\begin{aligned} x_i &= 0 \quad \forall i \in N \\ x_i &= p_i \quad \forall i \in \mathcal{B} \end{aligned}$$

Ed il valore della funzione obiettivo con tale BFS è  $z_0$ .

**Osservazione 11** Se  $\mathbf{r} \leq \mathbf{0}$  la soluzione è ottimale ed il valore massimizzato della funzione obiettivo è  $z_0$ .

**Lemma 2** Per ogni base ammissibile  $\mathcal{B}$ , le componenti del tableau sono univocamente definite ed assumono i seguenti valori

$$Q = -A_{\mathcal{B}}^{-1} \cdot A_N \tag{2.38}$$

$$\mathbf{p} = A_{\mathcal{B}}^{-1} \mathbf{b} \tag{2.39}$$

$$z_0 = \mathbf{c}_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} \mathbf{b} \tag{2.40}$$

$$\mathbf{r} = \mathbf{c}_N - (\mathbf{c}_{\mathcal{B}}^T A_{\mathcal{B}}^{-1} A_N)^T \tag{2.41}$$

*Dimostrazione* : Essendo che

$$\mathbf{x}_{\mathcal{B}} = [x_i : i \in \mathcal{B}]^T$$

$$\mathbf{x}_N = [x_i : i \notin \mathcal{B}]^T$$

La seguente identità è verificata

$$A\mathbf{x} = A_{\mathcal{B}}\mathbf{x}_{\mathcal{B}} + A_N\mathbf{x}_N = \mathbf{b}$$

quindi

$$A_{\mathcal{B}}\mathbf{x}_{\mathcal{B}} = \mathbf{b} - A_N\mathbf{x}_N$$

Essendo che  $A_B$  è per definizione non singolare (essendo  $B$  una base ammissibile), esiste l'inverso

$$\mathbf{x}_B = A_B^{-1} \mathbf{b} - A_B^{-1} \cdot A_N \mathbf{x}_N \quad (2.42)$$

Da qui

- $A_B^{-1} \mathbf{b} = \mathbf{p}$
- $-A_B^{-1} \cdot A_N = Q$

inoltre

$$z = \mathbf{c}^T \mathbf{x} = \mathbf{c}_B^T \mathbf{x}_B + \mathbf{c}_N^T \mathbf{x}_N$$

sostituendo  $\mathbf{x}_B$  come nell'equazione 2.42 si ha

$$z = \mathbf{c}_B^T (A_B^{-1} \mathbf{b} - A_B^{-1} \cdot A_N \mathbf{x}_N) + \mathbf{c}_N^T \mathbf{x}_N$$

con alcune semplici manipolazioni algebriche:

$$z = \mathbf{c}_B^T A_B^{-1} \mathbf{b} + (\mathbf{c}_N - \mathbf{c}_B^T A_B^{-1} A_N) \mathbf{x}_N$$

Da qui

- $z_0 = \mathbf{c}_B^T A_B^{-1} \mathbf{b}$
- $\mathbf{r}^T = \mathbf{c}_N - \mathbf{c}_B^T A_B^{-1} A_N$

■

Si osservi come il metodo del simplesso richiede una base ammissibile dalla quale partire, dato un LP in forma standard, si può trasformare in forma d'equazione aggiungendo le variabili slack, la matrice originale verrà affiancata da una matrice identità di  $m$  colonne ( $m$  numero originale di vincoli), queste sono banalmente linearmente indipendenti, quindi quella composta dalle variabili slack è sempre una valida base dalla quale partire.

Come procedere nel caso generale di un problema che si trova già in forma di equazione?

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

*Nota* : Si può assumere che  $\mathbf{b}$  abbia ogni componente positiva, se così non dovesse essere, il vincolo

$$a_{1,j}x_1 + \dots + a_{n,j}x_n = b_j$$

si potrebbe trasformare in

$$-a_{1,j}x_1 - \dots - a_{n,j}x_n = -b_j$$

Al problema originale si possono aggiungere delle variabili slack per ogni vincolo

$$A' \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = [A | \text{Id}_m] \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ y_1 \\ \vdots \\ y_m \end{bmatrix} = \mathbf{b} \quad \begin{aligned} \mathbf{x} &\geq 0 \\ \mathbf{y} &\geq 0 \end{aligned}$$

In tal modo si ha una BFS semplice associata ad una base contenente le variabili slack.

**Osservazione 12** *Se esiste una soluzione per  $A' \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{b}$  con le variabili slack uguali a zero, allora esiste una soluzione per il problema originale.*

Si può fare risolvendo il seguente LP con una nuova funzione obiettivo

$$\begin{aligned} \max & -y_1 - y_2 \cdots - y_m \\ A' \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} &= \mathbf{b} \\ \mathbf{x}, \mathbf{y} &\geq 0 \end{aligned} \quad (2.43)$$

chiaramente dato  $\mathbf{y} \geq 0$  il valore minimo che può assumere  $\mathbf{y}$  è zero. Quindi se il nuovo problema 2.43 ha soluzione ottimale diversa da zero, allora non esiste una soluzione per il problema originale.

Il ragionamento appena espresso porta al seguente risultato.

**Proposizione 11** *Il sistema di equazioni*

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

*ha soluzione se e solo se il seguente programma lineare*

$$\begin{aligned} \max & -y_1 - y_2 \cdots - y_m \\ A' \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} &= \mathbf{b} \\ \mathbf{x}, \mathbf{y} &\geq 0 \end{aligned}$$

*ha come valore ottimale 0.*

Nelle sezioni precedenti abbiamo visto come la ricerca di soluzioni ammissibili riguarda i punti contenuti in un poliedro (definito dai vincoli lineari), un problema correlato ma sostanzialmente diverso, riguarda lo stabilire se un dato poliedro sia vuoto o no. Si potrebbe ipotizzare, che tale problema sia simile alla risoluzione di un programma lineare, ma più semplice, in quanto un LP riguarda sia la ricerca di una soluzione ammissibile (punti nel poliedro), sia la ricerca del punto specifico che massimizza la funzione obiettivo. La seguente proposizione enuncia che tale ipotesi è falsa.

**Proposizione 12** *Sia  $\phi$  una funzione oracolo, che dato un poliedro definito da un sistema di disequazioni lineari, assume valore 1 se e solo se il poliedro non è vuoto (il sistema ha soluzione), altrimenti, assume valore 0. Se  $\phi$  esiste, è possibile risolvere un qualsiasi programma lineare tramite un numero finito di valutazioni di  $\phi$ .*

Piuttosto che una dimostrazione rigorosa sarà data una prova informale, sia

$$\begin{aligned} \max & \mathbf{c}^T \mathbf{x} \\ A\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

il programma lineare in questione, si assume che gli elementi di  $A$  e di  $\mathbf{c}$  siano numeri razionali e non reali. Si costruisce, dato un arbitrario  $\alpha \in \mathbb{Q}$  il seguente poliedro

$$P_\alpha = \begin{cases} \mathbf{c}^T \mathbf{x} \leq \alpha \\ A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{cases}$$

Si utilizza l'oracolo  $\phi$ , se  $P_\alpha$  è non vuoto, ossia se  $\phi(P_\alpha) = 1$ , allora il valore massimizzato del programma lineare è maggiore o uguale ad  $\alpha$ , altrimenti, se  $\phi(P_\alpha) = 0$ , il valore è minore. Si può fare ricerca binaria sui valori di  $\alpha$ , data l'assunzione iniziale sul fatto che i numeri coinvolti siano razionali, la ricerca termina in un numero finito di passi.

## 2.5 Soluzioni Degenere

Alcune BFS sono problematiche quando valutate durante il metodo del simplesso e possono portare ad una situazione di stallo in cui si considerano in maniera ciclica le stesse basi a ripetizione.

**Definizione 23** Dato un LP, sia  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  una BFS, se l'insieme

$$\{x_i \text{ t.c. } x_i > 0\}$$

ha cardinalità minore di  $n - m$  (alcune variabili di base sono uguali a zero), allora  $\mathbf{x}$  è una soluzione *degenere*.

Si consideri il seguente esempio

$$\begin{aligned} \max & x_2 \\ & x_1 \leq 1 \\ & -x_1 + x_2 \leq 0 \\ & x_1, x_2 \geq 0 \end{aligned} \quad (2.44)$$

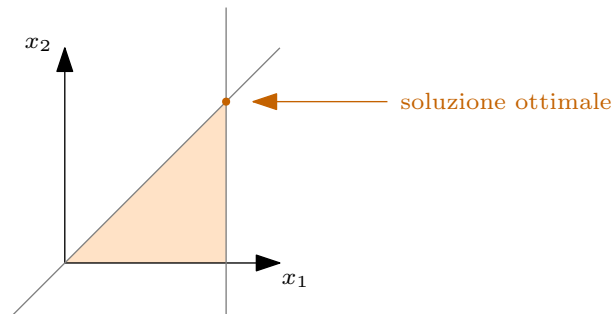


Figura 2.14: Poliedro delle soluzioni ammissibili

La soluzione ottimale è (banalmente)  $\bar{\mathbf{x}} = [1, 1]^T$ , il problema in forma di equazione assume la seguente forma

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (2.45)$$

La base di partenza è quella composta dalle variabili slack, ossia  $\mathcal{B} = \{3, 4\}$ , ponendo  $x_1, x_2 = 0$  si risolve per  $x_3, x_4$  trovando la BFS

$$[0 \ 0 \ 1 \ 0]^T$$

Questa è per definizione degenere, il tableau per  $\mathcal{B}$  è il seguente

$x_3 = 1 - x_1$
$x_4 = 0 + x_1 - x_2$
$z = 0 + x_2$

si osservi come c'è un'unica possibilità di sostituzione per il pivot, si sostituisce  $x_2$  con  $x_4$ , la nuova base è  $\mathcal{B} = \{2, 3\}$

$x_2 = 0 + x_1 - x_4$
$x_3 = 1 - x_1$
$z = 0 + x_1 - x_4$

La BFS in questo caso è identica, si noti come lo scambio delle variabili non ha fatto incrementare il valore di  $z$ , questa è la conseguenza di una soluzione degenere. In questo caso, scambiando  $x_1$  con  $x_3$  si ottiene la soluzione ottimale. Si noti che una BFS degenere può essere associata a più di una base (ciò non si verifica in ogni caso).