# Ricerca Operativa

## Costruzione di modelli di ottimizzazione: concetti generali

Prof. Paolo Brandimarte
Dip. di Scienze Matematiche – Politecnico di Torino
e-mail: `paolo.brandimarte@polito.it`
URL: `staff.polito.it/paolo.brandimarte`

Questa versione: 19 febbraio 2023

# Contenuto

---

Le slide seguenti sono tratte dal capitolo 2 di: P. Brandimarte, *Ottimizzazione per la Ricerca Operativa*, CLUT 2022.

- Uso delle variabili binarie nella programmazione lineare:
  - Alcuni modelli standard
  - Formulazioni basate su colonne
  - Vincoli logici
  - Funzioni lineari a tratti
- Struttura dei costi

quando la produzione o l'acquisto di un bene avvengono in quantità discrete o multipli di un lotto standard. Si distinguono casi interi puri (ILP – *Integer Linear Programming*) o misti-interi (MILP – *Mixed-Integer Linear Programming*).

Tuttavia, è molto più comune l'uso di variabili binarie, $x \in \{0, 1\}$, che rappresentano decisioni logiche.

Trattiamo i seguenti esempi:

- il modello dello zaino (knapsack);
- i modelli di set covering, set partitioning e set packing;
- i modelli basati su colonne;
- la rappresentazione di relazioni e vincoli logici;
- il collegamento di variabili binarie e continue mediante il trucco della grande $M$;
- la rappresentazione di funzioni lineari a tratti.

Sono dati un insieme di $n$ opportunità di investimento, ciascuna delle quali richiede un capitale iniziale $w_j$ e garantisce un ricavo (payoff) $p_j$, $j \in [n]$, e un budget $B$.

Occorre allocare il budget in modo da massimizzare il payoff complessivo.

Introduciamo variabili binarie, una per ogni oggetto da selezionare:

$$x_j = \begin{cases} 1 & \text{se l'oggetto } j \in [n] \text{ è selezionato,} \\ 0 & \text{altrimenti.} \end{cases}$$

Otteniamo quindi un problema di programmazione lineare binaria pura:

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} p_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} w_j x_j \le B \\ & x_j \in \{0, 1\}. \end{aligned}$$

Dati: un insieme $\mathcal{I}$ di $m$ oggetti e una famiglia $\mathcal{J}$ di $n$ sottoinsiemi di tali oggetti, $S_j \subset \mathcal{I}$, $j \in [n]$, descritti da una *matrice di incidenza* $\mathbf{A} \in \mathbb{R}^{m \times n}$:

$$a_{ij} = \begin{cases} 1 & \text{se l'oggetto } i \text{ è incluso nel sottoinsieme } j \\ 0 & \text{altrimenti,} \end{cases}$$

Problema di set covering: ogni sottoinsieme ha un costo $c_j$, e si vuole trovare una collezione di sottoinsiemi di minimo costo totale, in modo tale che ogni oggetto sia incluso in almeno uno dei sottoinsiemi.

Se associamo ai sottoinsiemi variabili binarie $x_j$ poste a 1 se il sottoinsieme viene selezionato, il modello risultante è:

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq 1, \qquad i \in [m] \\ & x_j \in \{0, 1\}. \end{aligned}$$

Esso può essere espresso in forma matriciale come

$$\begin{aligned} \min \quad & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{1}_m \\ & \mathbf{x} \in \{0,1\}^n, \end{aligned}$$

dove $\mathbf{1}_m$ indica un vettore colonna di $m$ elementi tutti posti a uno.

Avremo un problema di set partitioning se il vincolo è $\mathbf{A}\mathbf{x} = \mathbf{1}_m$, e un problema di set packing se si impone $\mathbf{A}\mathbf{x} \leq \mathbf{1}_m$.

Nel caso del packing è naturale cambiare il senso della funzione obiettivo e risolvere un problema di massimizzazione.

I modelli di set covering e partitioning sono alla base di diversi modelli adatti a gestire situazioni complesse, in cui la soluzione deve sottostare a molti vincoli piuttosto complessi da modellare.

Si usano spesso approcci che vanno sotto il nome di *column generation* (vedi VRP), in cui le colonne vengono generate dinamicamente in modo che dipende dal problema.

Consideriamo un problema di staffing, in cui dobbiamo stabilire quante casse aprire, per ogni fascia oraria, in un ipermercato.

Per formulare un modello didattico, supponiamo che ogni giorno consista di 8 ore, $t = 1, 2, \ldots, 8$, e che ci sia un fabbisogno di personale variabile nel tempo, rappresentato da $R_t$, $t = 1, \ldots, 8$.

Supponiamo che ogni turno consista di 4 ore consecutive, secondo uno dei pattern: (lavoro, pausa, lavoro, lavoro); (lavoro, lavoro, pausa, lavoro). I turni possono iniziare nei periodi $t = 1, 2, 3, 4, 5$, e vogliamo soddisfare il fabbisogno al minimo costo.

Possiamo introdurre un set di colonne costituite, il cui elemento in posizione $t \in [8]$ è 1 se l'ora corrispondente è coperta dal turno, 0 altrimenti:

$$\begin{bmatrix} 1\\1\\0\\1\\0\\0\\0\\0 \end{bmatrix} \begin{bmatrix} 1\\0\\1\\1\\0\\0\\0\\0 \end{bmatrix} \begin{bmatrix} 0\\1\\1\\0\\1\\0\\0\\0 \end{bmatrix} \begin{bmatrix} 0\\1\\0\\1\\1\\0\\0\\0 \end{bmatrix} \begin{bmatrix} 0\\0\\1\\1\\0\\1\\0\\0 \end{bmatrix} \begin{bmatrix} 0\\0\\1\\0\\1\\1\\0\\0 \end{bmatrix} \begin{bmatrix} 0\\0\\0\\1\\1\\0\\1\\0 \end{bmatrix} \begin{bmatrix} 0\\0\\0\\1\\0\\1\\1\\0 \end{bmatrix} \begin{bmatrix} 0\\0\\0\\0\\1\\1\\0\\1 \end{bmatrix} \begin{bmatrix} 0\\0\\0\\0\\1\\0\\1\\1 \end{bmatrix}$$

Otteniamo dieci colonne $\mathbf{a}_j \in \{0,1\}^8$, $j = 1, \ldots, 10$, che possono essere raggruppate in una matrice binaria $\mathbf{A} \in \{0,1\}^{8\times 10}$, dove l'elemento $a_{tj}$ è 1 se il lavoratore che segue il pattern $j$ è attivo durante l'ora $t$, 0 altrimenti. Se indichiamo con $x_j \in \mathbb{Z}_+$ il numero (intero non negativo) di lavoratori impiegati secondo lo schema $j$, il numero di lavoratori attivi nell'ora $t$ è

$$\sum_{j=1}^{10} a_{tj} x_j,$$

e non deve essere inferiore al fabbisogno $R_t$.

Se assumiamo che tutti i pattern abbiano lo stesso costo, l'obiettivo di minimizzare i costi di staffing coincide con la minimizzazione del numero di assunti:

$$\begin{aligned}
\min \quad & \sum_{j=1}^{10} x_j \\
\text{s.t.} \quad & \sum_{j=1}^{10} a_{tj} x_j \geq R_t, && t = 1, \ldots, 8 \\
& x_j \in \mathbb{Z}_+, && j = 1, \ldots, 10.
\end{aligned}$$

Per esempio, possiamo associare variabili binarie $x_a$ e $x_b$ a condizioni logiche $a$ e $b$ e imporre condizioni di tipo booleano:

- $a$ or $b$ (or inclusivo): $x_a + x_b \geq 1$
- $a$ xor $b$ (or esclusivo): $x_a + x_b = 1$
- se $a$ allora $b$ ($a \Rightarrow b$): $x_a \leq x_b$
- $a$ se e solo se $b$ ($a \Leftrightarrow b$): $x_a = x_b$

Il not si traduce nel complemento a 1 di una variabile: $1 - x_a$.

Un po' più complesso è il caso $c = (a \text{ and } b)$. La scrittura $x_c = x_a x_b$ introduce una non linearità, che può però essere eliminata imponendo:

$$x_c \leq x_a; \quad x_c \leq x_b; \quad x_c \geq x_a + x_b - 1. \tag{1}$$

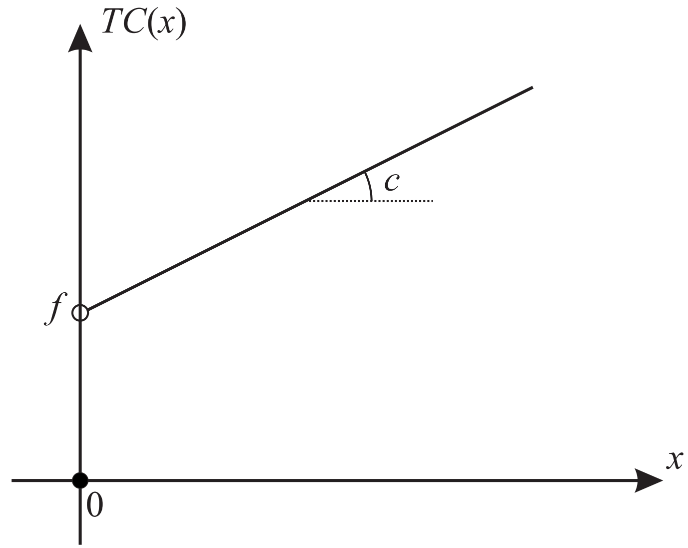
In linea di principio, potremmo introdurre una funzione gradino

$$\gamma(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x = 0 \end{cases}$$

e rappresentare il costo totale come

$$\text{TC}(x) = cx + f\gamma(x). \qquad (2)$$

La funzione gradino $\gamma$ è una funzione discontinua e non lineare.




Per rappresentare la struttura di costo variabile più fisso in modo lineare, possiamo introdurre una variabile binaria

$$\delta = \begin{cases} 1 & \text{se } x > 0, \\ 0 & \text{altrimenti.} \end{cases}$$

ed esprimere il costo totale come

$$\text{TC}(x) = cx + f\delta.$$

Per legare $x$ e $\delta$ introduciamo una costante $M$ sufficientemente grande:

$$x \leq M\delta. \qquad (3)$$

Questo tipo di vincolo, noto come trucco della grande $M$, è molto flessibile, ma:

- In realtà, la rappresentazione non è perfettamente equivalente a quella dell'equazione (2), perché è possibile pagare il costo fisso, ponendo $\delta = 1$ a poi mantenere $x = 0$. In pratica, questo non avviene nella soluzione ottima.

- Dal punto di vista computazionale, vincoli del tipo (3) risultano problematici in quanto tendono a rendere inefficaci le strategie di enumerazione implicita per la soluzione di modelli MILP. È bene cercare di trovare il più piccolo upper bound sul livello della attività $x$.

- Sono da evitare espressioni come $x\delta$, che introducono una non linearità. Tale regola ha qualche eccezione per problemi a struttura speciale, quando si vuole evitare l'impatto dei vincoli con grande $M$ sull'efficienza degli algoritmi enumerativi.

**Vincolo di cardinalità.** dato un insieme di variabili decisionali, desideriamo limitare il numero di quelle che assumono un valore strettamente positivo.

Consideriamo quindi un insieme di variabili non negative $x_j$, $j \in \mathcal{J}$, e un limite $K <\mid \mathcal{J} \mid$ sulla cardinalità. Basta introdurre delle variabili binarie $\delta_j$ corrispondenti alle $x_j$,

$$\delta_j = \begin{cases} 1 & \text{se } x_j > 0,\ j \in \mathcal{J} \\ 0 & \text{altrimenti,} \end{cases}$$

e imporre il vincolo

$$\sum_{j \in \mathcal{J}} \delta_j \leq K. \tag{4}$$

**Variabili semi-continue.** Supponiamo che il livello di una attività, se essa viene intrapresa, deve essere compreso tra un limite inferiore $m_i$ e uno superiore $M_i$. Questo *non* corrisponde al vincolo convesso $m_i \leq x_i \leq M_i$, ma a quello non convesso $x_i \in \{0\} \cup [m_i, M_i]$.

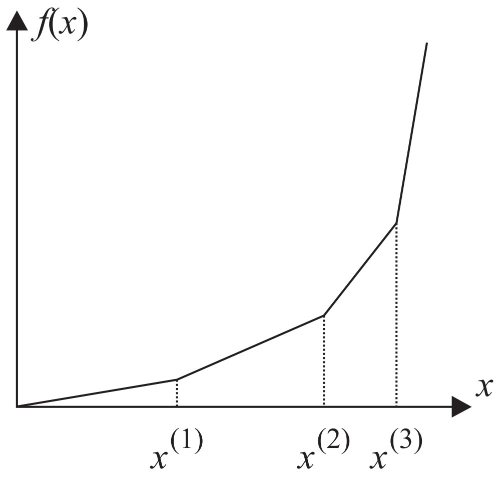
Usando il trucco della grande $M$ possiamo esprimere il vincolo in forma lineare:

$$x_i \geq m_i \delta_i, \qquad x_i \leq M_i \delta_i$$

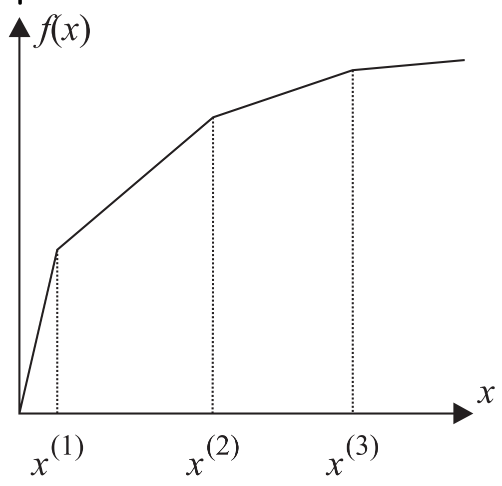
Si pone spesso il problema di rappresentare relazioni non lineari tra variabili o costi che dipendono da economie (o diseconomie) di scala:

- Il costo totale di trasporto può dipendere in modo non lineare dal volume della merce trasportata in un intervallo di tempo.
- Il costo di acquisto di un bene può dipendere in modo non lineare da sconti sulla quantità.
- Il costo di transazione sui mercati finanziari può dipendere in modo non lineare dall'ammontare venduto o comprato, e dalla liquidità dell'asset.

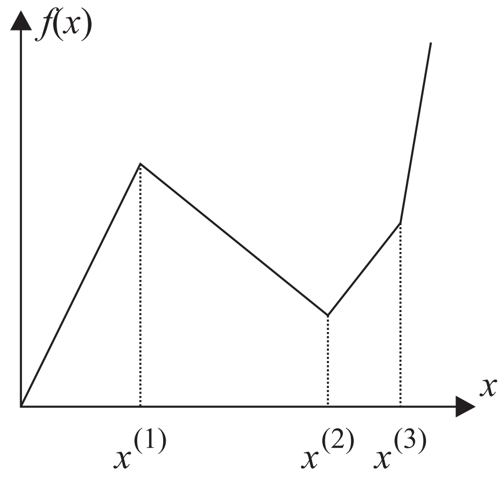
Può essere utile ricorrere ad approssimazioni lineari a tratti



(a)



(b)

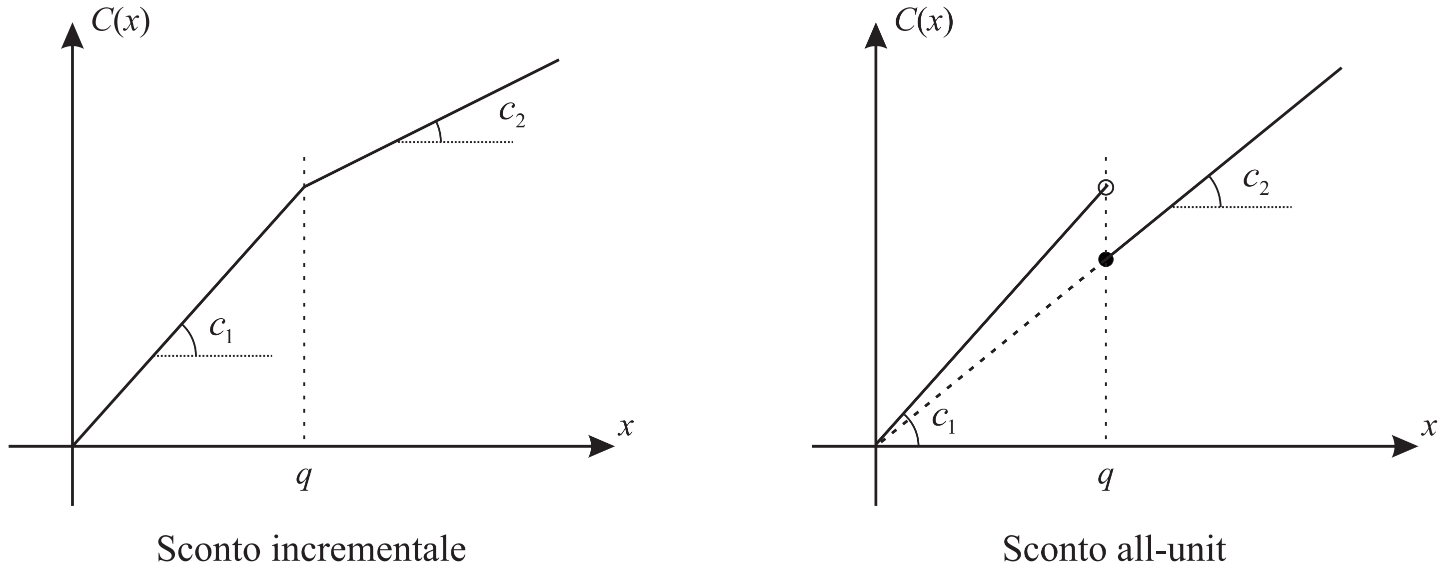


(c)

A volte il costo unitario viene ridotto quando la quantità supera una certa soglia critica $q$:

1. nel caso *all-unit* lo sconto si applica a tutte le unità acquistate;

2. nel caso *incrementale* lo sconto si applica solo alla quota eccedente il valore di soglia.



Per descrivere una funzione lineare a tratti si possono seguire due approcci diversi.

- Fornire la lista dei breakpoint sull'asse delle ascisse e i relativi coefficienti angolari. Se la funzione passa per l'origine (come da figura precedente), potremmo elencare i breakpoint, $(x^{(1)}, x^{(2)}, x^{(3)})$, e la pendenza della funzione su ciascun intervallo, $(c_1, c_2, c_3, c_4)$.

- Fornire la lista dei punti estremi estremi di ciascun segmento, in termini di coppie breakpoint–valore $(x^{(i)}, y^{(i)})$, dove $y^{(i)} = f(x^{(i)})$.

Conoscendo i breakpoint e le pendenze, possiamo rappresentare una funzione continua $f(x)$ come

$$f(x) = \begin{cases} c_1 x, & 0 \le x \le x^{(1)} \\ c_1 x^{(1)} + c_2 \left(x - x^{(1)}\right), & x^{(1)} \le x \le x^{(2)} \\ c_1 x^{(1)} + c_2 \left(x^{(2)} - x^{(1)}\right) + c_3 \left(x - x^{(2)}\right), & x^{(2)} \le x \le x^{(3)} \\ c_1 x^{(1)} + c_2 \left(x^{(2)} - x^{(1)}\right) + c_3 \left(x^{(3)} - x^{(2)}\right) & \\ \qquad\qquad + c_4 \left(x - x^{(3)}\right), & x^{(3)} \le x \end{cases}$$

Avremo quindi casi diversi in funzione dei coefficienti angolari.

- Se $c_1 < c_2 < c_3 < c_4$ (costi marginali crescenti), allora $f(x)$ è convessa, come in figura (a).

- Se $c_1 > c_2 > c_3 > c_4$ (costi marginali decrescenti), allora la funzione è concava, come in figura (b).

- Nel caso di pendenze $c_i$ arbitrarie la funzione non è né concava né convessa, come in figura (c).

Il caso convesso è facile da trattare introducendo un insieme di variabili ausiliarie, una per ogni intervallo,

$$x = z_1 + z_2 + z_3 + z_4, \tag{5}$$

mentre i contributi sono vincolati dalla lunghezza di ciascun intervallo:

$$\begin{aligned}
&0 \le z_1 \le x^{(1)} \\
&0 \le z_2 \le \left(x^{(2)} - x^{(1)}\right) \\
&0 \le z_3 \le \left(x^{(3)} - x^{(2)}\right) \\
&0 \le z_4.
\end{aligned}$$

Su tale base, la funzione $f(x)$ può essere rappresentata come

$$f(x) = c_1 z_1 + c_2 z_2 + c_3 z_3 + c_4 z_4.$$

Il meccanismo funzione perché nella minimizzazione di un costo convesso le variabili vengono attivate nella sequenza corretta. Non c'è incentivo ad attivare $z_2$, più costosa di $z_1$, a meno che quest'ultima non sia saturata al suo livello superiore.

Al contrario, la cosa non funziona nel caso concavo. Se $c_2 < c_1$, il solver attiverà $z_2$ prima di $z_1$.

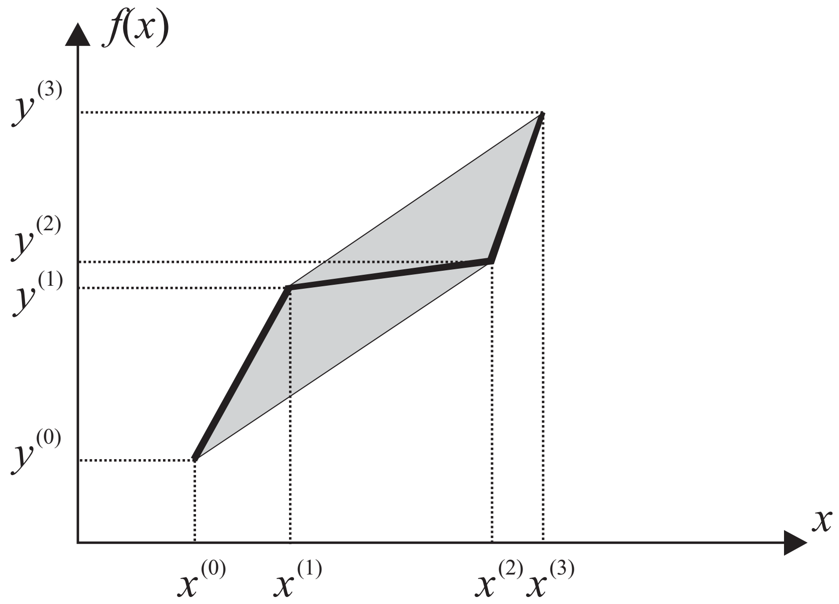
Un esempio concreto è il caso degli sconti incrementali: non possiamo ottenere lo sconto se acquistiamo una quantità inferiore al valore di soglia minima che lo attiva.

Possiamo allora scambiare una non convessità per un'altra, introducendo variabili decisionali binarie.

Rappresentiamo la funzione mediante un set di nodi $(x^{(i)}, y^{(i)})$, $i = 0, 1, 2, 3$. Se consideriamo il segmento che unisce i nodi $(x^{(i)}, y^{(i)})$ e $(x^{(i+1)}, y^{(i+1)})$, possiamo descriverlo come la combinazione convessa dei due punti estremi:

$$x = \lambda x^{(i)} + (1 - \lambda) x^{(i+1)},$$
$$y = \lambda y^{(i)} + (1 - \lambda) y^{(i+1)},$$

dove $0 \leq \lambda \leq 1$.



Per rappresentare la funzione su tutto l'intervallo di interesse, potremmo considerare combinazioni convesse

$$x = \sum_{i=0}^{3} \lambda_i x^{(i)},\ y = \sum_{i=0}^{3} \lambda_i y^{(i)}, \qquad \text{dove} \quad \sum_{i=0}^{3} \lambda_i = 1,\ \lambda_i \geq 0. \tag{6}$$

Tuttavia, così facendo costruiamo la ricopertura convessa dei quattro punti nel piano, che è il poliedro in grigio.

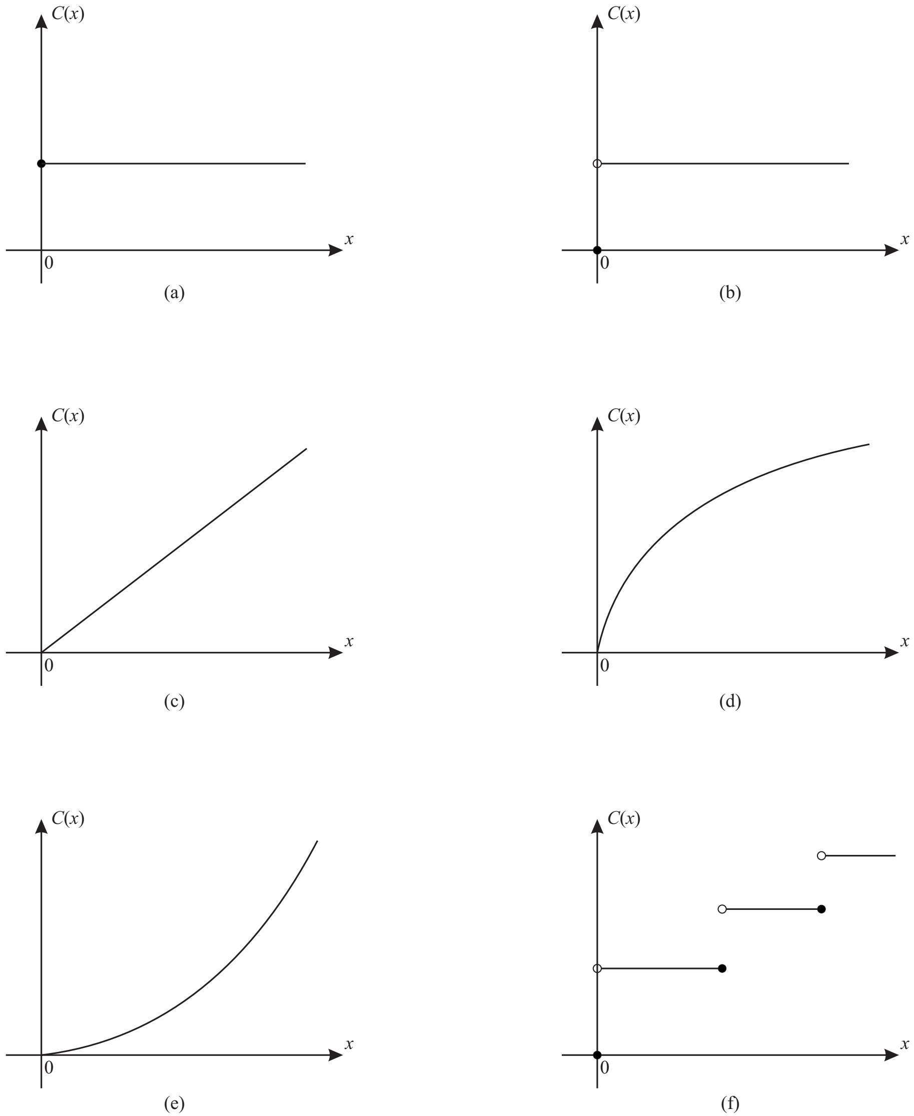
Per considerare solo coppie di nodi adiacenti, introducendo una variabile binaria $\delta_i$ per ogni intervallo $[x^{(i-1)}, x^{(i)}]$, $i = 1, 2, 3$.

Ai vincoli dell'equazione (6) dobbiamo quindi aggiungere

$$\begin{aligned}
&0 \leq \lambda_0 \leq \delta_1 \\
&0 \leq \lambda_1 \leq \delta_1 + \delta_2 \\
&0 \leq \lambda_2 \leq \delta_2 + \delta_3 \\
&0 \leq \lambda_3 \leq \delta_3 \\
&\sum_{i=1}^{3} \delta_i = 1, \qquad \delta_i \in \{0, 1\}.
\end{aligned}$$

In effetti, se $\delta_1 = 1$ possiamo combinare solo i nodi $i = 0$ e $i = 1$, cioè rappresentiamo il primo segmento. Se $\delta_2 = 1$ possiamo combinare solo i nodi $i = 1$ e $i = 2$ associati al secondo e così via.

(a) **Costo fisso affondato** (*fixed/sunk cost*), che rappresenta un costo di struttura che si paga comunque.
(b) **Costo fisso** o **tariffa fissa**, che si differenzia dal caso precedente in quanto se $x = 0$ il costo è zero.
(c) **Costo variabile lineare/proporzionale** (*linear/proportional cost*), in cui il costo varia linearmente, con una pendenza costante, in funzione del livello dell'attività.
(d) **Economia di scala**, nel qual caso il costo è una funzione non lineare concava del livello $x$.
(e) **Diseconomia di scala**, nel qual caso il costo è una funzione non lineare convessa del livello $x$.
(f) **Costi semivariabili**, nel qual caso i costi sono costanti a tratti.



Dal punto di vista pratico, le strutture possono essere combinate nei modi più svariati, ed è importante notare come la natura di un costo possa dipendere dal livello decisionale.

Per esempio, in un problema tattico come il mix di produzione, il costo della capacità produttiva è un dato, e quindi è un costo affondato.

Salendo a livello strategico, in un problema di pianificazione della capacità produttiva il costo della capacità è invece oggetto della decisione.

Si può dire che, salendo di livello gerarchico, tutti i costi sono variabili o semivariabili.

La funzione $C(x)$ rappresenta il contributo al costo totale di una attività. Oltre al costo totale, possono essere utili altre due misure legate al costo (come pure al ricavo):

- Il **costo medio**, $\overline{C}(x) \doteq C(x)/x$.

- Il **costo marginale**, $C'(x)$.

In un esempio precedente abbiamo ottenuto il prezzo ottimale, quando la domanda è funzione del prezzo e si intende massimizzare il ricavo.

Consideriamo il caso in cui si ha una funzione di domanda inversa che fornisce il prezzo $p(q)$ a cui possiamo vendere una quantità $q$ di un bene.

Assumiamo che l'obiettivo sia la massimizzazione del profitto $\pi(q)$:

$$\pi(q) = p(q) \cdot q - c(q) = r(q) - c(q),$$

dove la funzione di costo $c(q)$ è assunta differenziabile, e introduciamo la funzione ricavo $r(q) \doteq p(q) \cdot q$, pure essa assunta differenziabile.

Se la funzione profitto è concava, possiamo applicare la condizione di ottimalità

$$\pi'(q^*) = r'(q^*) - c'(q^*) = 0 \qquad \Rightarrow \qquad r'(q^*) = c'(q^*).$$

Osserviamo che all'ottimo sono uguali ricavo e costo marginale.

Se il ricavo marginale eccede il costo marginale, l'incremento di ricavo all'aumentare di $q$ è maggiore dell'incremento del costo, per cui conviene aumentare $q$.

Al contrario, se il ricavo marginale è minore del costo marginale, conviene ridurre la produzione.