

Gestione delle Scorte

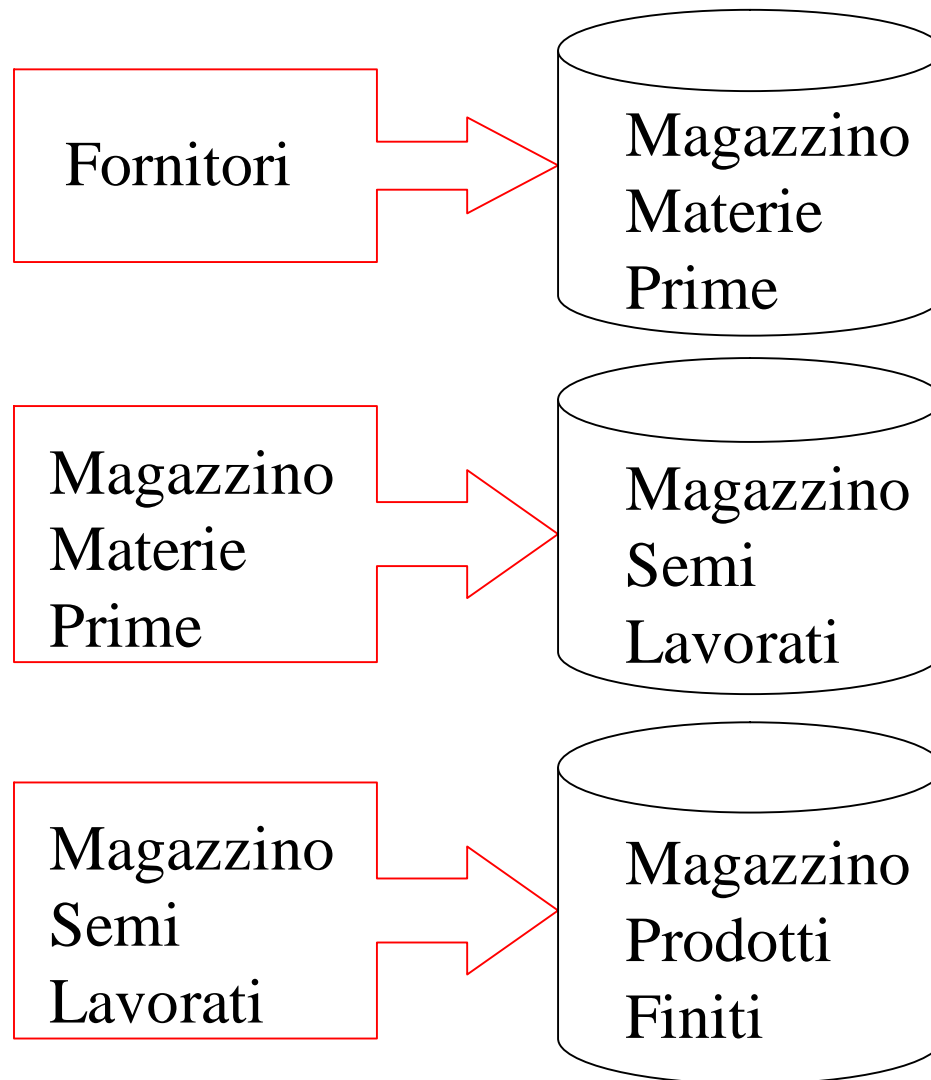
Renato Bruni

bruni@dis.uniroma1.it

Il materiale presentato è derivato da quello dei proff. A. Sassano e C. Mannino

Gestione delle Scorte (Inventory Management)

Produzione vista come processo di Approvvigionamento del Magazzino



Le Scorte sono Utili

1. Economie di Scala

Produzione superiore al fabbisogno immediato

- Minimizza l'incidenza dei costi fissi
- Favorisce l'apprendimento
- Offre la possibilità di ottenere sconti

Necessità di accumulare il “surplus” produttivo

2. Flessibilità

- Disaccoppiamento delle fasi produttive
- Fuori servizio di macchinari
- Carenza di materie prime sul mercato

3. Equipartizione della Produzione

- Produzione in periodi di domanda scarsa

Ma di Più non è necessariamente Meglio

4. Speculazione

- Produzione quando si prevede che un bene aumenterà di prezzo

1. Le scorte rappresentano però un immobilizzazione di risorse

2. Causano Spese

- Gestione del Magazzino
- Sorveglianza del Magazzino
- Possibile Deterioramento, Furto, etc.

3. Tentativi di Riduzione

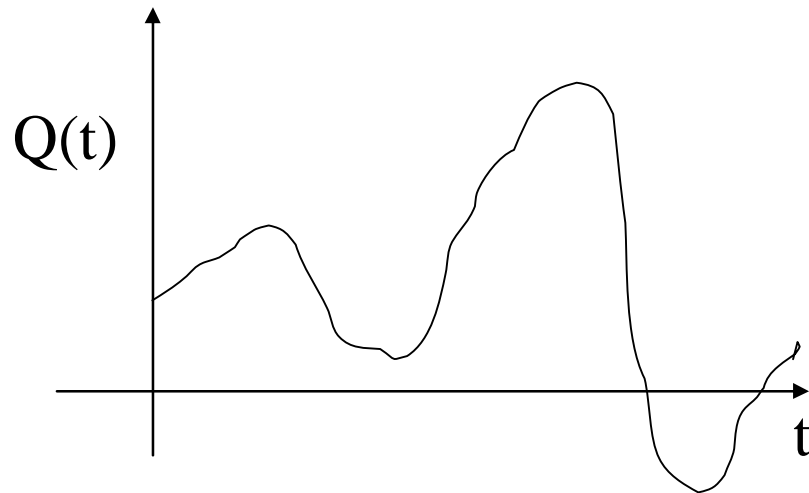
- Stockless Production
- Just In Time

Bisogna gestire le scorte opportunamente

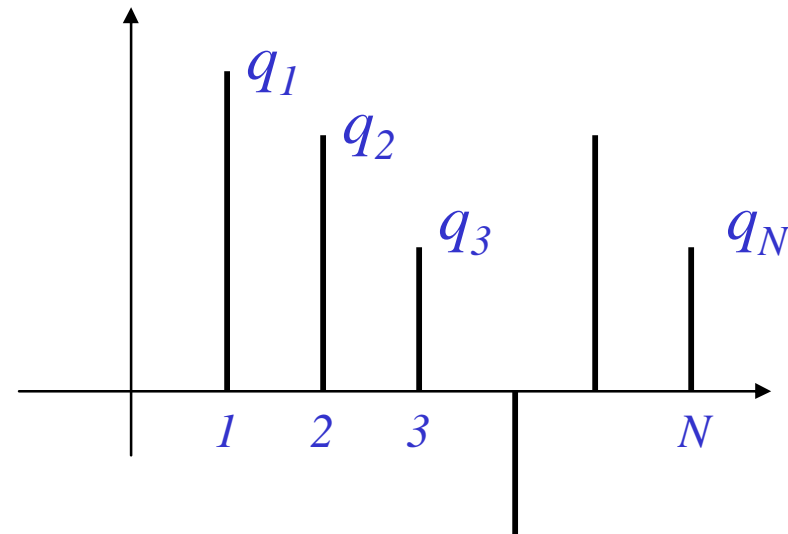
Controllo Continuo vs Discreto

OBIETTIVO

Determinare l'andamento nel tempo delle giacenze di magazzino (scorte) che massimizza un'opportuna funzione di utilità.



Controllo Continuo



(1,2,3,...,N) Periodi di Controllo

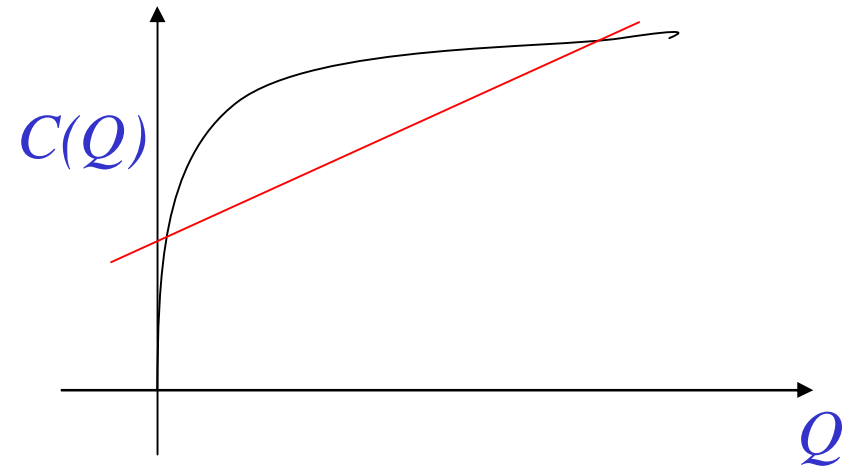
Controllo Discreto

Andamento del Costo Variabile di Produzione

PARAMETRI FONDAMENTALI

1. Costo di Produzione: $C(Q)$

Funzione concava (per le economie di scala) della quantità prodotta



Esempi:

(a)

Approvvigionamento da Fornitori

Costo di acquisto + Costo di trasporto

(b)

Approvvigionamento da Magazzino Semilavorati

Costo del Semi-lavorato

Contabilità Industriale o Analitica

Costo Fisso di Produzione

PARAMETRI FONDAMENTALI

2. *Costo Fisso di Produzione*: A

Costo sostenuto per *attivare* la produzione

Indipendente dalle quantità prodotte

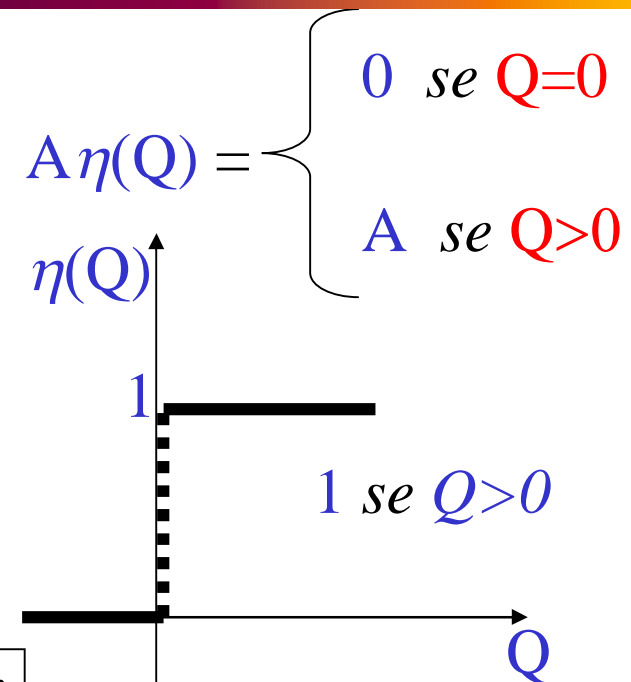
Esempi:

(a) *Approvvigionamento da Fornitori*

- Costo amministrativo dell'Ordine
- Quantità minima ordinabile

(b) *Produzione effettiva*

- Costi di "setup" e "changeover"



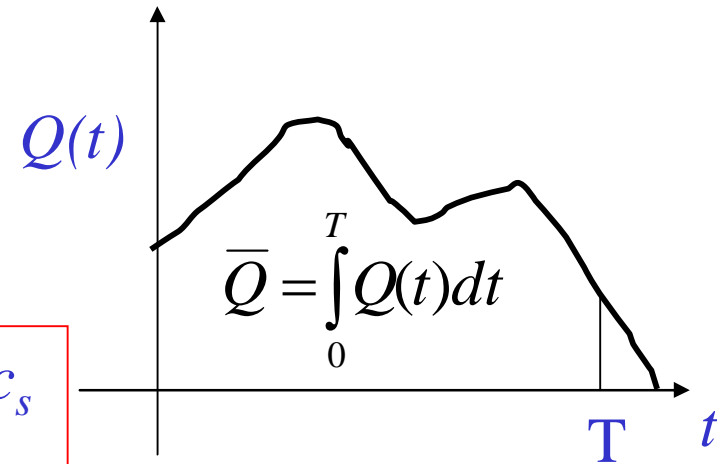
Andamento del Costo Variabile di Stoccaggio

PARAMETRI FONDAMENTALI

3. Costo di Stoccaggio: $h(Q)$

$$h(Q) = c_s(Q) \times \bar{Q}$$

Prodotto del costo unitario di stoccaggio c_s
per la giacenza complessiva \bar{Q}



$c_s(Q)$:

Funzione del Costo/Opportunita' del valore monetario dei beni immagazzinati (quanto mi avrebbe reso quel capitale immobilizzato in magazzino?)

Per semplicita' lo calcolo con: *Minimo Tasso di Interesse Accettabile* r_0

$$c_s(Q) = r_0 \times C(Q)/Q + \dots$$

Costo unitario di produzione quando la quantita' e' Q

Costo Fisso di Stoccaggio

PARAMETRI FONDAMENTALI

4. *Costo Fisso di Stoccaggio:* π

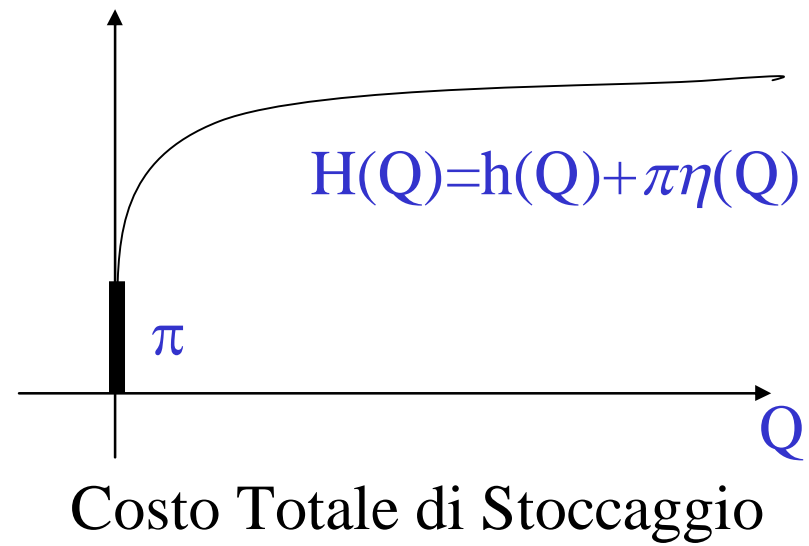
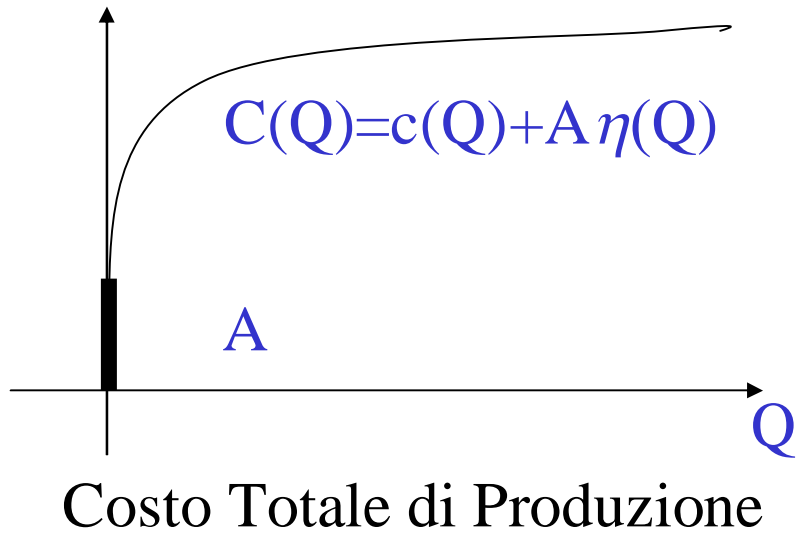
Costo sostenuto per le scorte indipendentemente dalle quantità immagazzinate

Dovuto a:

$$\pi \eta(Q) = \begin{cases} 0 & \text{se } Q=0 \\ \pi & \text{se } Q>0 \end{cases}$$

- Costo Fisso di Gestione (ad es. salari degli addetti, etc.)

Costi Complessivi



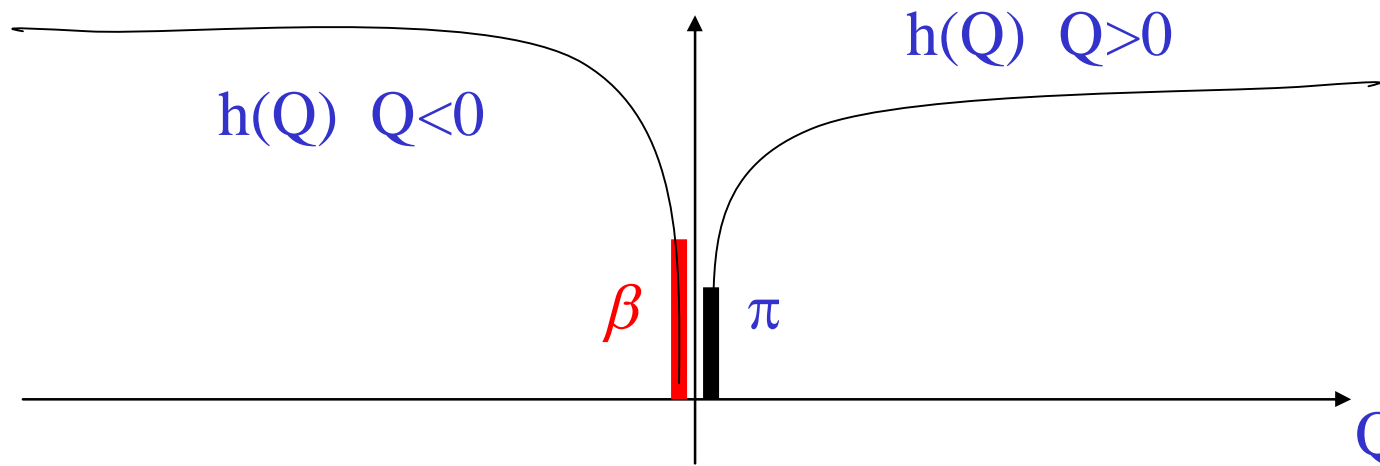
Caso con Backlogging

Backlogging: soddisfare la domanda attuale con una produzione futura (= quantità negativa in magazzino, quantità già ordinata)

- *Questo causa un costo* (scontento del cliente, gestione, etc.)

5. Costo variabile di “Backlogging”: $h(Q)$ quando $Q < 0$

6. Costo Fisso di “Backlogging”: β



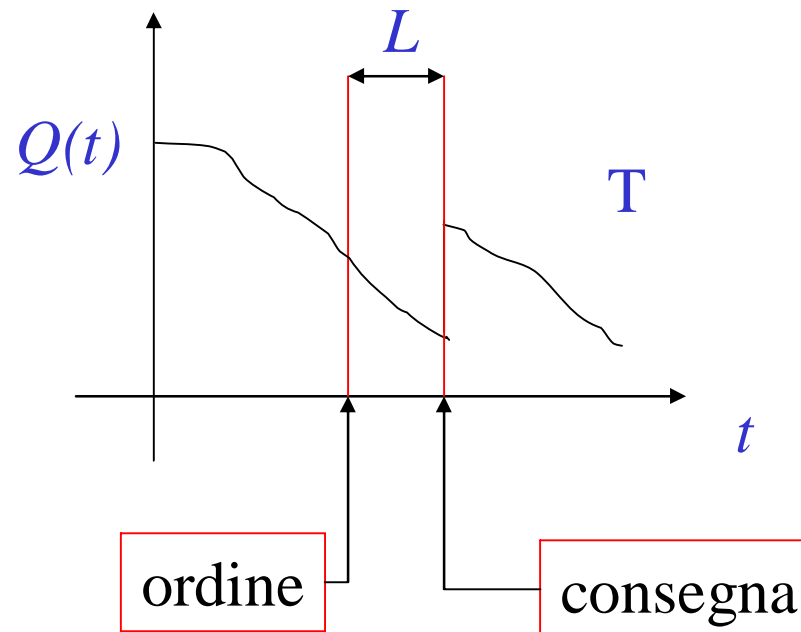
Costo Totale di Stoccaggio: $H(Q) = h(Q) + \pi\eta(Q) + \beta\eta(-Q)$

Lead Time

ALTRI PARAMETRI CARATTERISTICI

7. “Lead Time” L

- Tempo intercorrente tra l’ordine e la consegna

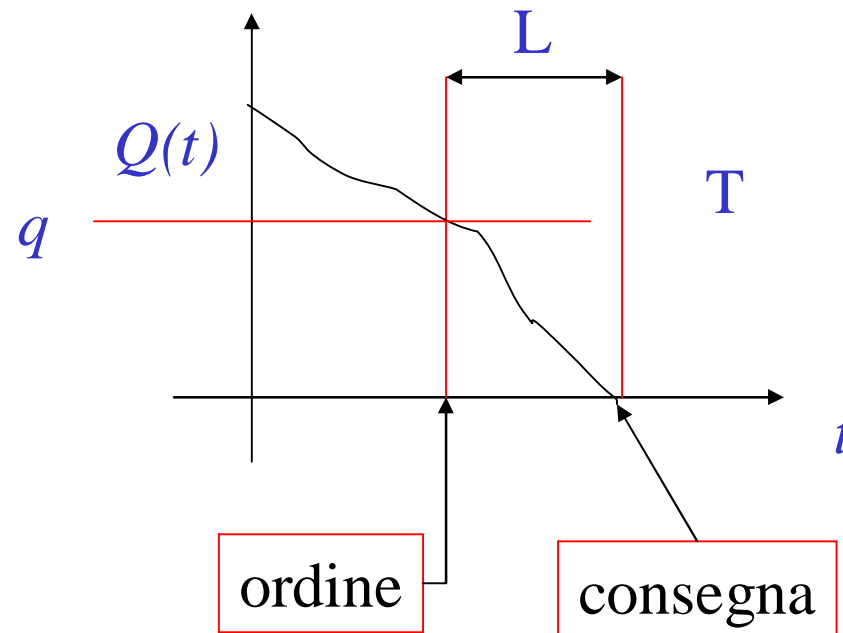


Scorta Minima

ALTRI PARAMETRI CARATTERISTICI

8. Scorta Minima q

- Giacenza minima di magazzino
ovvero: *quantita' consumata durante il "Lead Time"*



Vari Modelli Possibili

CARATTERISTICHE DEI MODELLI (1)

1. Natura dei Parametri (domanda, costi, “lead time”)

Deterministica: Parametri conosciuti con certezza

Aleatoria (Stocastica): Parametri rappresentati da variabili aleatorie con opportune distribuzioni

2. Struttura della domanda e della produzione

- Costante nel tempo (modelli classici)
- Variabile

Controllo continuo: il modello classico (EOQ)

Modello Classico - Ipotesi

- Modello Deterministico
- Controllo Continuo
- Singolo bene

- Domanda *uniforme* D (D unità per anno, $D/12$ al mese, ...)

- *Costo di produzione*: $C(Q) = A\eta(Q) + vQ$
 - Costo fisso di produzione
 - Costo unitario

- *Costo di stoccaggio*: $h(Q) = r_0 \overline{Q} \frac{C(Q)}{Q}$
 - ↓ MARR
 - Giacenza complessiva

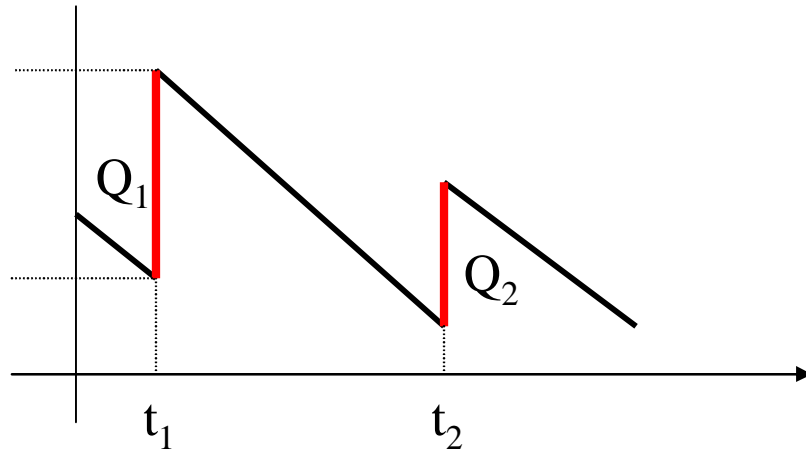
- *Costo fisso di stoccaggio nullo*
- “*Lead time*” *nullo* (il bene prodotto è immediatamente disponibile)
- “*Backlogging*” *non consentito*

Modello Classico - Obiettivo

Vogliamo determinare:

1. Istanti di produzione (t_1, t_2, \dots)
2. Quantità Q_1, Q_2, \dots da produrre in ciascun istante di produzione

con **l'obiettivo** di soddisfare la domanda e minimizzare i costi di produzione e stoccaggio



Q_1 prodotto nell'istante t_1

Q_2 prodotto nell'istante t_2

Poichè i parametri non variano nel tempo, la quantità ottima da produrre sarà sempre la stessa (Q^*) in ogni istante e verrà detta lotto di produzione o lotto di riordino o “economic order quantity” (EOQ)

Gestione Scorte: Modello Classico

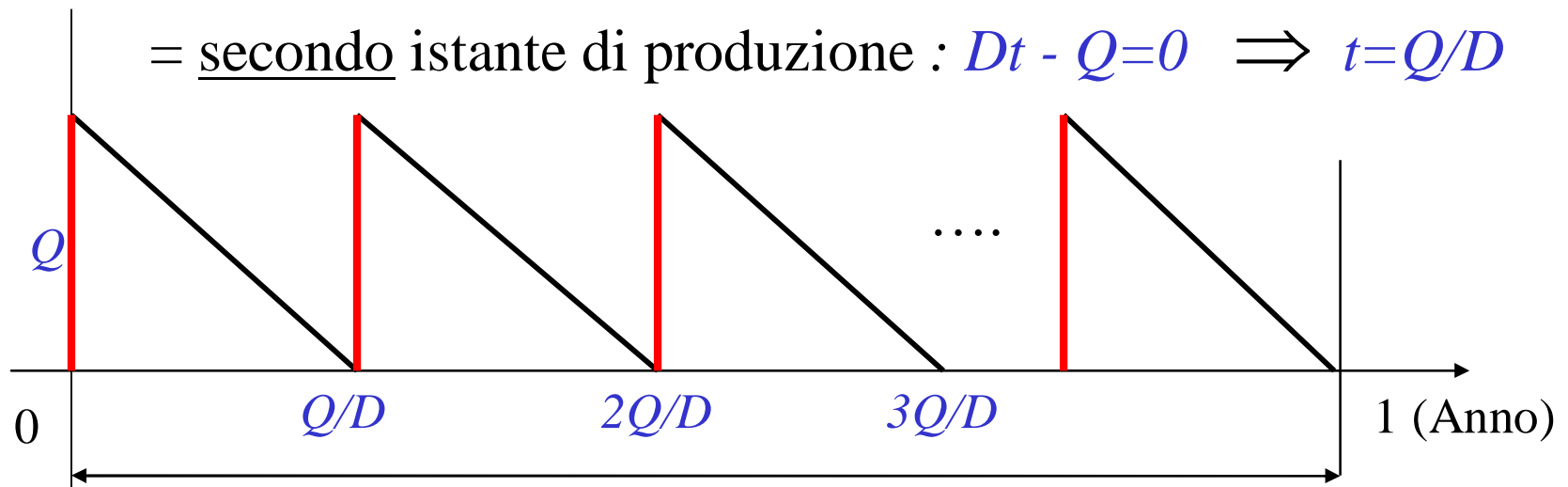
- Scorte nulle all'istante 0 (necessaria: *produzione all'istante 0*)
- Quantità prodotte uguali nei diversi istanti di produzione
- Domanda da servire fino all'istante t :

$$D(t) = Dt \quad \text{con } t \in [0, \dots, 1] \quad (\text{tasso di domanda costante})$$

D = domanda complessiva nell'orizzonte temporale $[0, \dots, 1]$

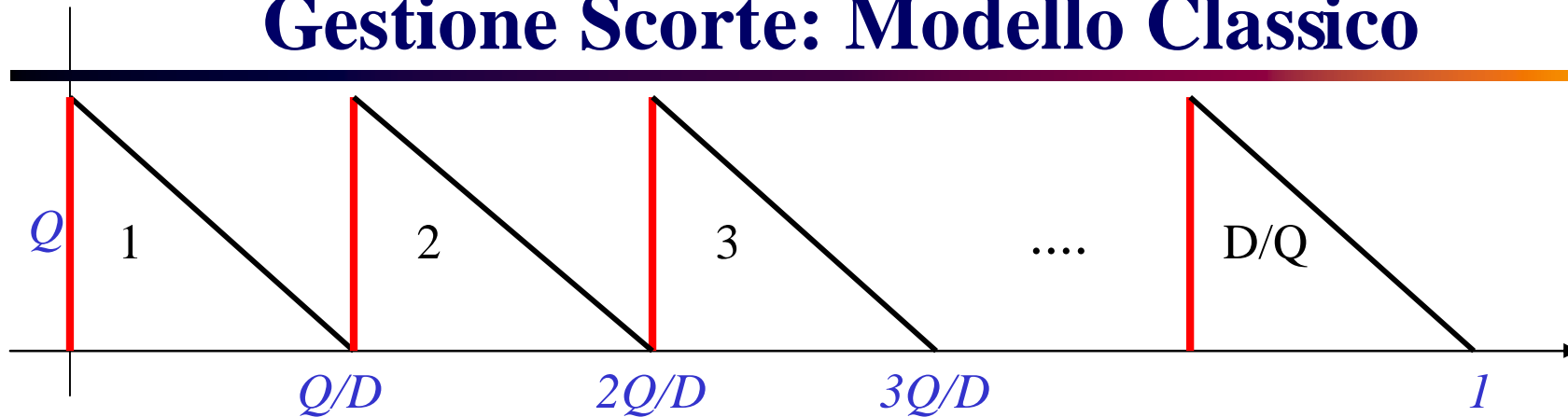
- Primo istante nel quale la scorta si annulla =

$$= \text{secondo istante di produzione : } Dt - Q = 0 \Rightarrow t = Q/D$$



- Numero istanti produttivi: D/Q

Gestione Scorte: Modello Classico



Costo di Produzione $C(Q) \times D/Q = (A + vQ) \times D/Q = \frac{AD}{Q} + vD$

Costo di Stoccaggio $h(Q) = r_0 \bar{Q} \frac{C(Q)}{Q} = r_0 \bar{Q} \left(\frac{A + vQ}{Q} \right) = r_0 \bar{Q} \left(\frac{A}{Q} + v \right)$

\bar{Q} è l'integrale della giacenza nell'intervallo $[0, \dots, 1]$

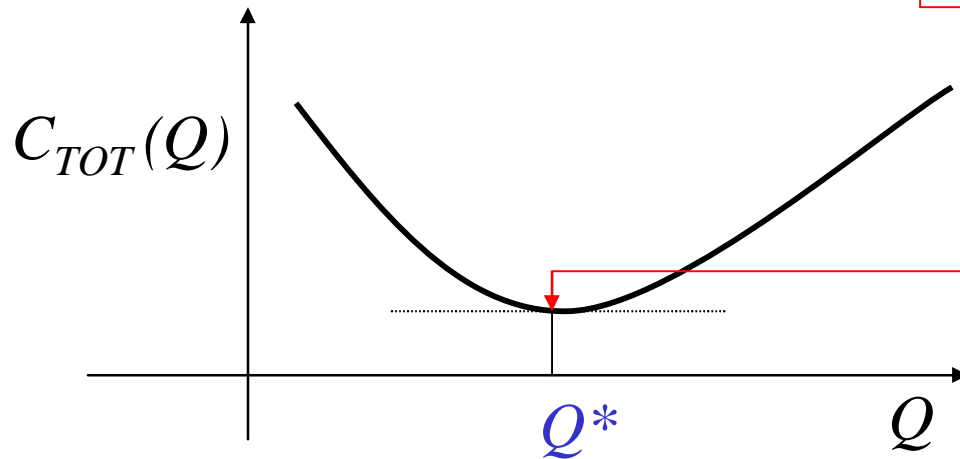
\bar{Q} è la somma delle aree dei triangoli, ovvero:

$$\bar{Q} = \frac{D}{Q} \frac{Q^2}{2D} = \frac{Q}{2}$$

Quindi: $h(Q) = r_0 \frac{Q}{2} \left(\frac{A}{Q} + v \right) = r_0 \left(\frac{A}{2} + \frac{vQ}{2} \right)$

Gestione Scorte: Modello Classico

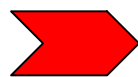
Costo Totale $C_{TOT}(Q) = h(Q) + C(Q) = \frac{AD}{Q} + vD + r_0 \left(\frac{A}{2} + \frac{vQ}{2} \right)$



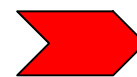
$$\frac{dC_{TOT}(Q)}{dQ} = 0$$

$$\frac{dC_{TOT}(Q)}{dQ} = -\frac{AD}{Q^2} + r_0 \frac{v}{2} = 0$$

$$-\frac{AD}{Q^2} + r_0 \frac{v}{2} = 0$$



$$Q^2 = \frac{2AD}{r_0 v}$$

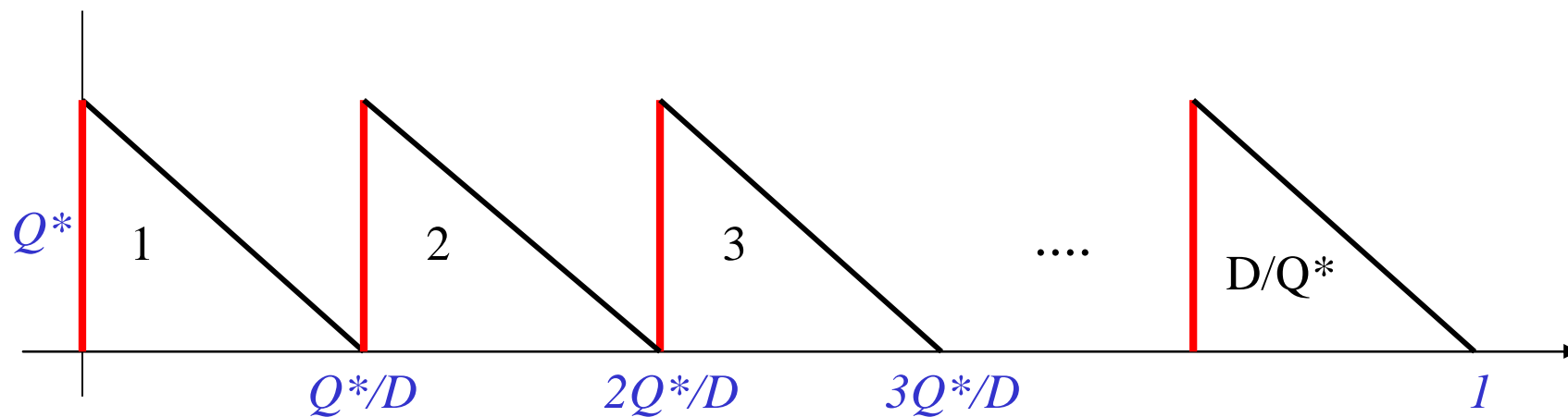


$$Q^* = \sqrt{\frac{2AD}{r_0 v}}$$

Gestione Scorte: Modello Classico

$$Q^* = \sqrt{\frac{2AD}{r_0 v}}$$

EOQ “Economic Order Quantity”
= Lotto di Riordino Ottimo



L'intervallo tra due ordini successivi è:

$$T_{EOQ} = \frac{Q^*}{D} = \sqrt{\frac{2AD}{r_0 v}} \frac{1}{D} = \sqrt{\frac{2A}{r_0 D v}}$$

Gestione Scorte: Modello Classico

$$Q^* = \sqrt{\frac{2AD}{r_0 v}}$$

$$T_{EOQ} = \sqrt{\frac{2A}{r_0 D v}}$$

ESEMPIO:

Domanda: $D = 2400$ scatole annue

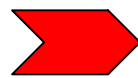
Costo unitario del bene: $v = 0.4$ Euro

Costo fisso di produzione: $A = 3.2$ Euro

MARR: $r_0 = 24\%$ (0.24)

$$Q^* = \sqrt{\frac{2AD}{r_0 v}} = \sqrt{\frac{2 \times 3.2 \times 2400}{0.24 \times 0.4}} = 400$$

$$T_{EOQ} = \sqrt{\frac{2A}{r_0 D v}} = \frac{1}{6}$$



2 mesi

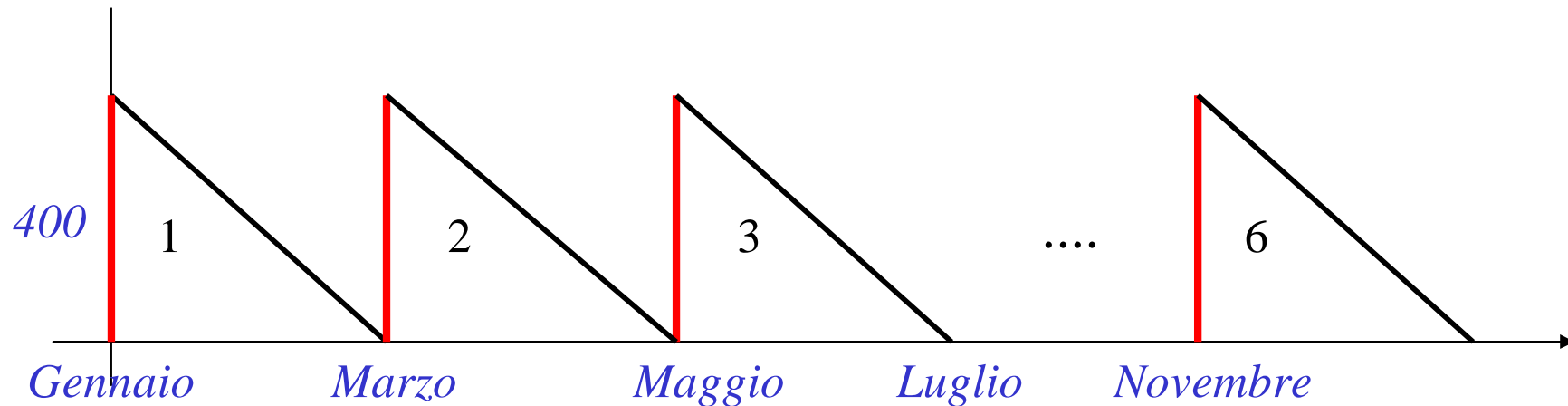
Gestione Scorte: Modello Classico

Domanda: $D = 2400$ scatole annue

Costo unitario del bene: $v = 0.4$ Euro

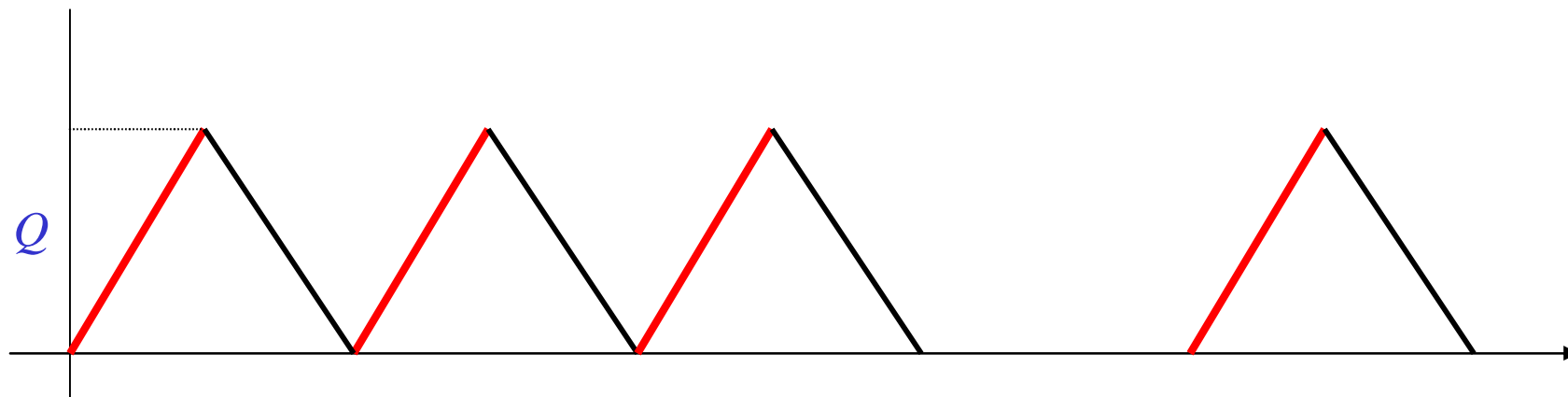
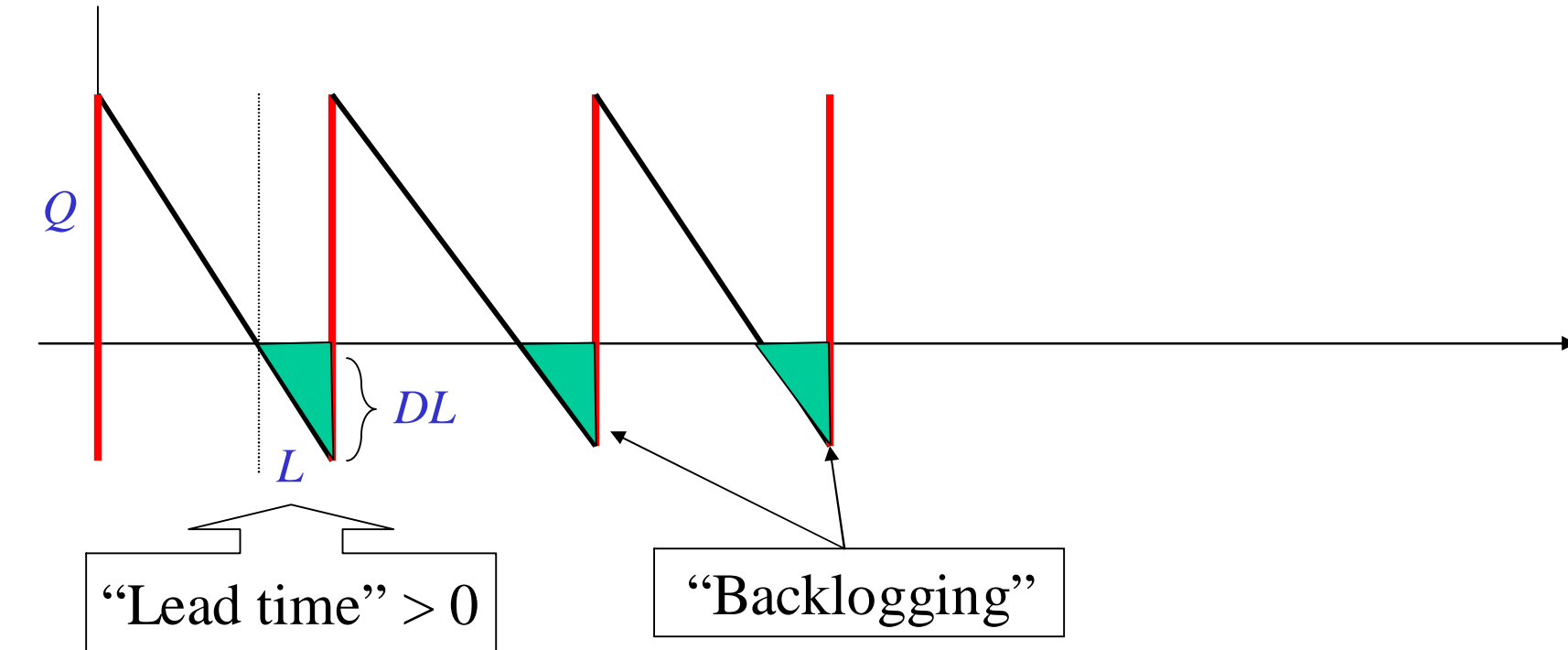
Costo fisso di produzione: $A = 3.2$ Euro

MARR: $r_0 = 24\%$ (0.24)



$$\begin{aligned} C_{TOT}(Q^*) &= \frac{AD}{Q^*} + vD + r_0 \left(\frac{A}{2} + \frac{vQ^*}{2} \right) = \\ &= 3.2 \times 6 + 960 + 0.24 (1.6 + 80) = \\ &= 979.2 + 19.58 \end{aligned}$$

Generalizzazioni del Modello Classico



Produzione non istantanea (tasso di produzione costante)

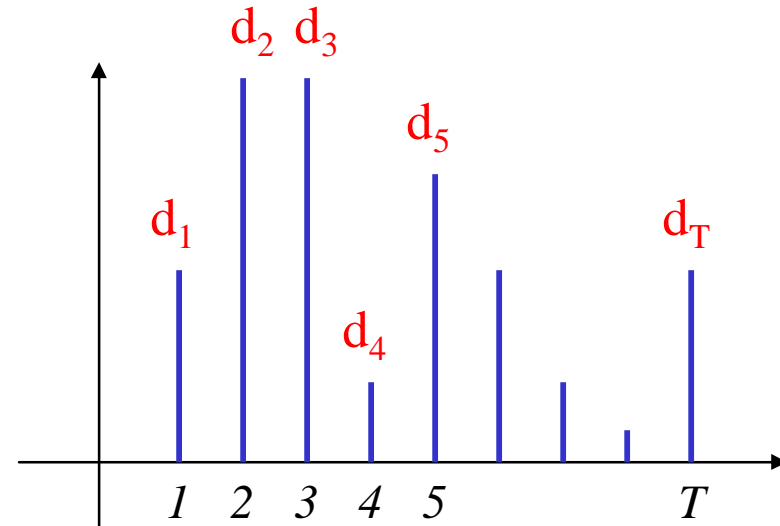
Controllo discreto e deterministico

Programmazione della Produzione - Ipotesi

- Modello Deterministico
- Controllo Discreto [$1, 2, 3, \dots, T$]
- Singolo bene

- Domanda Variabile nel tempo

$$[d_1, d_2, d_3, \dots, d_T] \quad d_i > 0$$



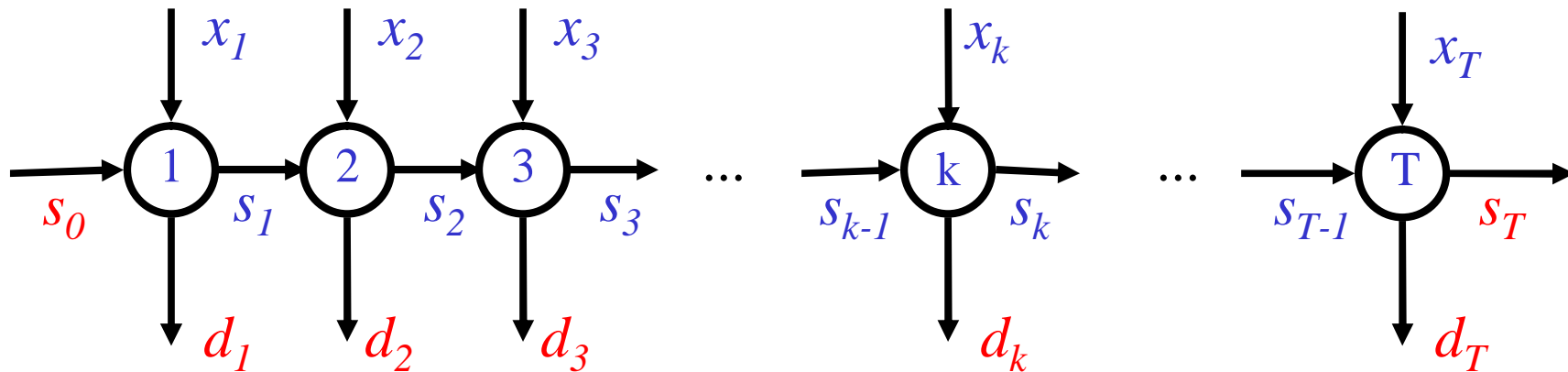
- Funzioni Costo (Produzione e Stoccaggio) Variabili nel tempo
(es: Produrre nel periodo k costa meno che nel periodo h)
- “*Lead time*” *nullo* (il bene prodotto è immediatamente disponibile)

Obiettivo

Vogliamo determinare:

1. Quantità x_1, x_2, \dots da produrre in ciascun periodo
2. Quantità s_1, s_2, \dots da immagazzinare in ciascun periodo

con **l'obiettivo** di soddisfare la domanda e minimizzare i costi di produzione e stoccaggio

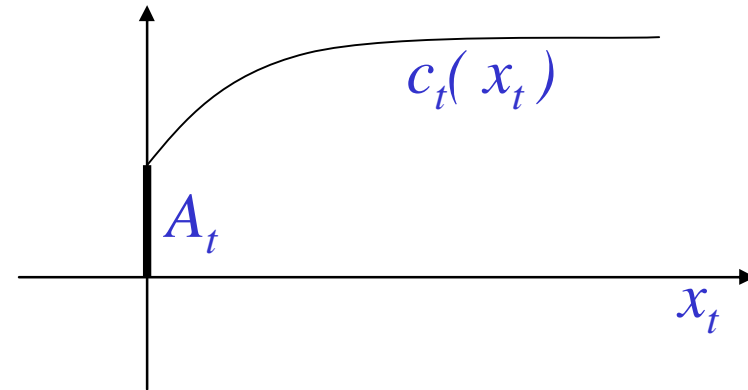


- s_0 Giacenza di magazzino all'inizio dell'orizzonte temporale
- s_T Giacenza di magazzino alla fine dell'orizzonte temporale

Andamento dei Costi

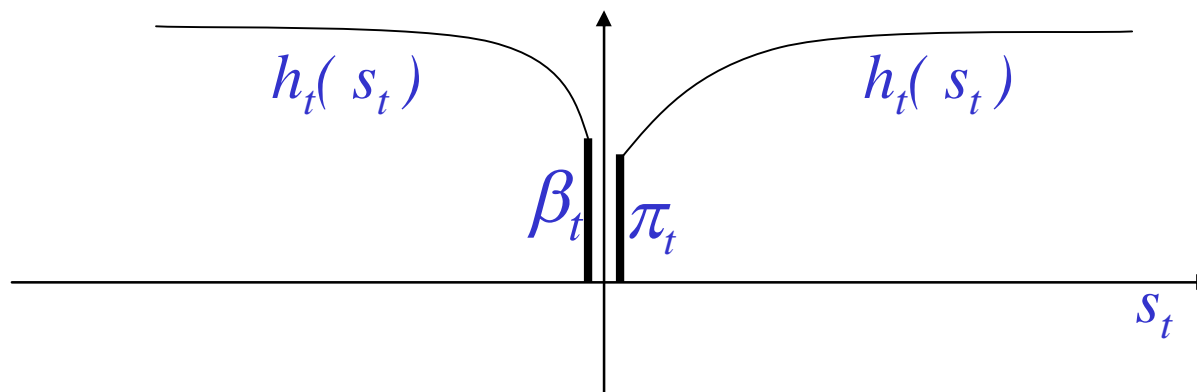
Costo di Produzione nel periodo t

$$C_t(x_t) = A_t \eta(x_t) + c_t(x_t)$$

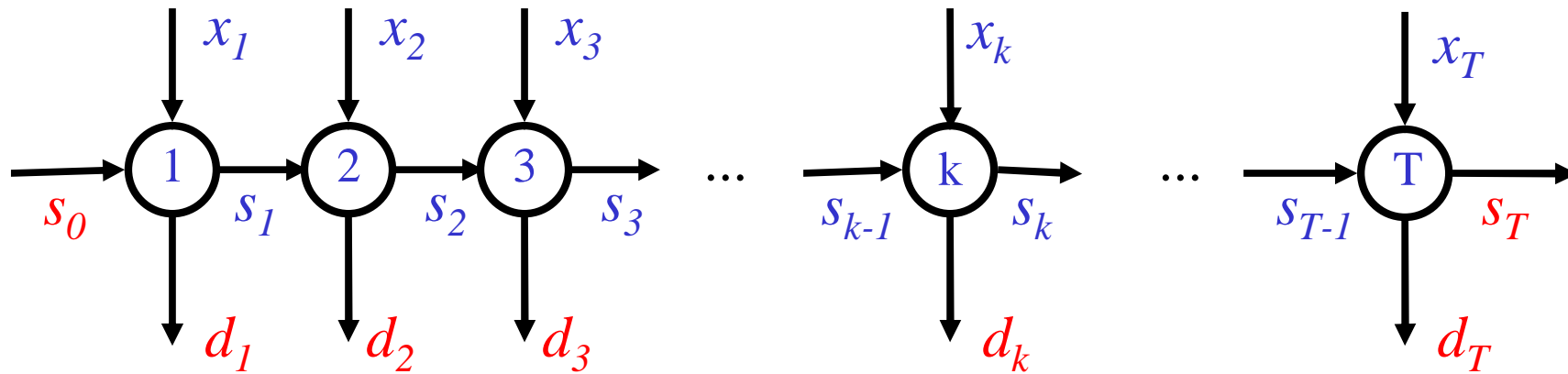


Costo di Stoccaggio nel periodo t

$$H_t(s_t) = \pi_t \eta(s_t) + \beta_t \eta(-s_t) + h_t(s_t)$$



Come Modellare il Problema?



$$\min f(x, s) = \sum_{t=1}^T (C_t(x_t) + H_t(s_t))$$

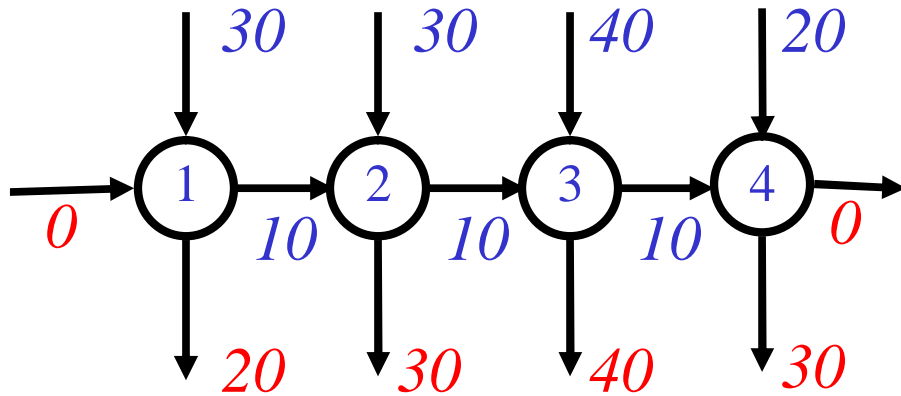
$$x_k + s_{k-1} - s_k = d_k \quad k = 1, \dots, T$$

$$x_k, s_k \geq 0 \quad k = 1, \dots, T$$

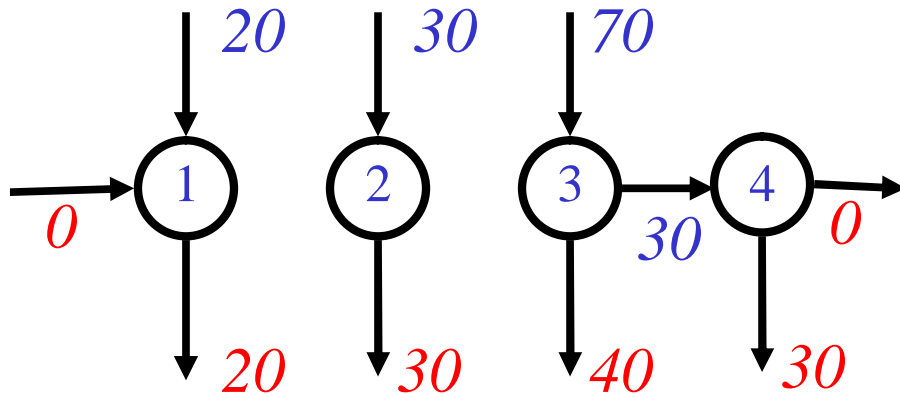
$$s_k \geq 0 = \text{“no backlogging”}$$

Tipi di Soluzioni

$$x_k + s_{k-1} - s_k = d_k \quad k = 1, \dots, T$$



Esempi di rappresentazioni.



Modello di Programmazione della Produzione

Assunzione. Scorte iniziali e finali nulle. $s_0 = 0, s_T = 0$.

- Le variabili s_0, s_T possono essere eliminate

$$\min f(x, s) = \sum_{t=1}^T (C_t(x_t) + H_t(s_t))$$

$$\left\{ \begin{array}{l} x_1 - s_1 = d_1 \\ x_k + s_{k-1} - s_k = d_k \quad k = 2, \dots, T-1 \\ x_T + s_{T-1} = d_T \\ x_k, s_k \geq 0 \quad k = 1, \dots, T-1 \\ x_T \geq 0 \end{array} \right.$$

$$s_k \geq 0 = \text{“no backlogging”}$$

OSS: T variabili di produzione x , $T-1$ variabili di scorta s

- T vincoli di domanda.
- $2T - 1$ variabili non negative
- (x, s) insieme di soluzioni di un sistema di disequazioni lineari \rightarrow poliedro.

Insieme delle Soluzioni Ammissibili

$$\left\{ \begin{array}{l} x_1 - s_1 = d_1 \\ x_k + s_{k-1} - s_k = d_k \quad k = 2, \dots, T-1 \\ x_T + s_{T-1} = d_T \\ x_k, s_k \geq 0 \quad k = 1, \dots, T-1 \\ x_T \geq 0 \end{array} \right.$$

Teorema. L'insieme di soluzioni ammissibili del problema di programmazione della produzione è un poliedro limitato (*politopo*)

Dim. Sommando tra loro i T vincoli di domanda si ottiene

$$\sum_{j=1}^T x_j = \sum_{j=1}^T d_j \quad (\text{la produzione totale eguaglia la domanda complessiva})$$

$$\longrightarrow x_i \leq \sum_{j=1}^T d_j = D \quad i = 1, \dots, T \quad (\text{dalla non-negatività della variabili } x)$$

Sommando tra loro i vincoli di domanda da $r > 1$ a T si ottiene

$$\sum_{j=r}^T x_j + s_{r-1} = \sum_{j=r}^T d_j \quad \longrightarrow \quad s_{r-1} = \sum_{j=r}^T d_j - \sum_{j=r}^T x_j \leq \sum_{j=r}^T d_j \leq D \quad r = 2, \dots, T$$



Oss. Da $\sum_{j=r}^T x_j + s_{r-1} = \sum_{j=r}^T d_j$ e $s_{r-1} \geq 0$ \longrightarrow $x_r \leq \sum_{j=r}^T d_j \quad r = 1, \dots, T$

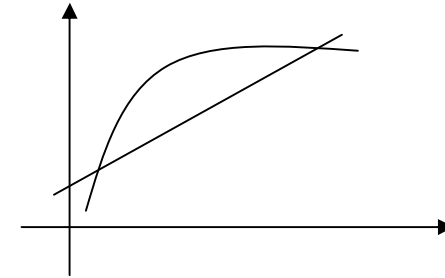
Struttura Funzione Obiettivo

- Problema di Programmazione Concava con Vincoli Lineari

$$\min f(x, s) \leftarrow \text{Funzione concava}$$

$$A \begin{pmatrix} x \\ s \end{pmatrix} = b \quad \left\{ \begin{array}{l} \text{A matrice } 2T-1 \times T \end{array} \right.$$

$$\begin{pmatrix} x \\ s \end{pmatrix} \geq 0_{2T-1} \quad \left\{ \begin{array}{l} \text{Forma standard} \end{array} \right.$$



$$\begin{pmatrix} x \\ s \end{pmatrix} = y$$

$$\begin{array}{l} \min f(y) \\ Ay = b \\ y \geq 0_{2T-1} \end{array}$$



Forma compatta

Proprietà funzione concava. $f: R^n \rightarrow R$ funzione concava, se e solo se, dati p punti $v_1, \dots, v_p \in R^n$ si ha che

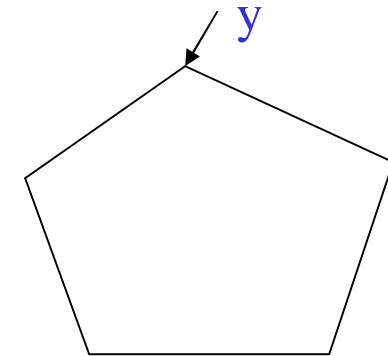
$$f\left(\sum_{k=1}^p u_k v_k\right) \geq \sum_{k=1}^p u_k f(v_k) \quad \text{per ogni } u_k \geq 0, k=1, \dots, p \quad \text{tali che } \sum_{k=1}^p u_k = 1$$

Vertici di un Poliedro

$$P = \{y \in R^{2T-1} : Ay = b, y \geq 0_{2T-1}\}$$

Politopo

y è un *vertice* di P se e solo se è una soluzione di base ammissibile (SBA)



- SBA definita da una sottomatrice quadrata nonsingolare B di A

$$By_B + Ny_N = b, y_B \geq 0_T, y_N \geq 0_{T-1}$$

$$\begin{cases} y_B = B^{-1}b \geq 0_T \\ y_N = 0_{T-1} \end{cases}$$

SBA definita da B

- Una SBA di P ha al più T componenti diverse da zero

Soluzioni che sono Vertici

Perchè i vertici (SBA) sono importanti?

Teorema: Un problema con funzione obiettivo concava $f(y)$ e regione ammissibile costituita da un politopo (non vuoto) ha sempre una soluzione ottima su un vertice.

Dimostrazione:

y^* soluzione ottima ($f(y^*) \leq f(y)$ per ogni $y \in P$)

$Ext(P) = \{v_1, v_2, \dots, v_p\}$ vertici di P (v^* : $f(v^*) \leq f(v)$ per ogni $v \in Ext(P)$)

$$y^* = \sum_{k=1}^p u_k v_k; \quad u_k \geq 0; \quad \sum_{k=1}^p u_k = 1$$

$$f(y^*) = f\left(\sum_{k=1}^p u_k v_k\right) \geq \sum_{k=1}^p u_k f(v_k) \geq \sum_{k=1}^p u_k f(v^*) = f(v^*)$$

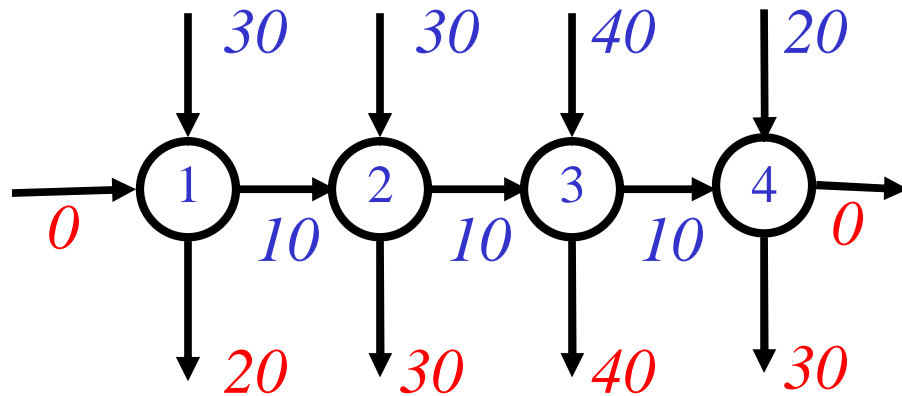
concavità

v^* soluzione ottima CVD.

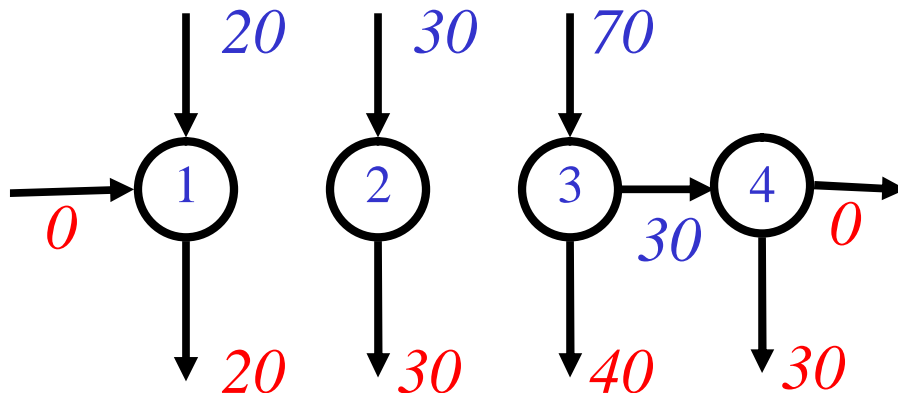
Tipi di Soluzioni

Possiamo limitare la ricerca della soluzione ottima alle SBA !

Come caratterizzare (e riconoscere) una SBA ?



Soluzione
ammissibile
non di base



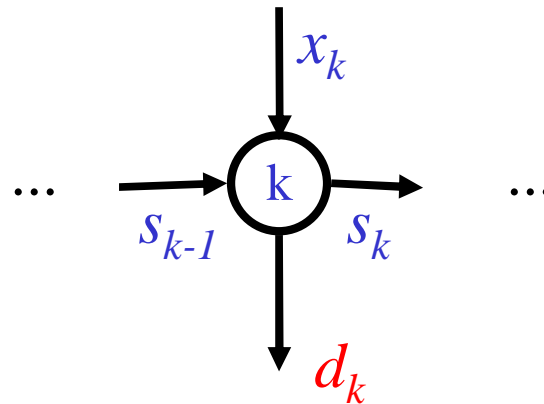
Soluzione
ammissibile
di base (SBA)

Distinguere le Soluzioni

$$\begin{cases} y_B = B^{-1}b \geq 0_T \\ y_N = 0_{T-1} \end{cases}$$

SBA definita da B

- Una SBA ha al più T componenti diverse da zero



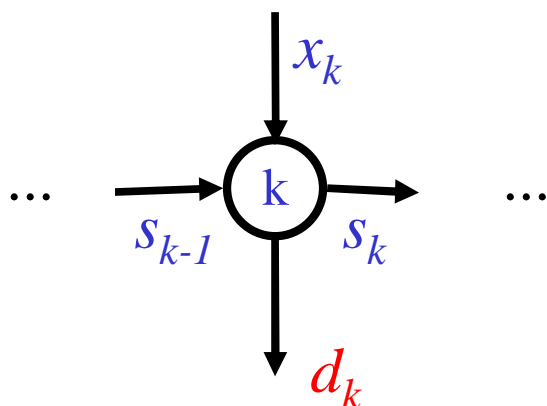
In ogni periodo k deve essere: $x_k + s_{k-1} > 0$

Una soluzione ha almeno T componenti diverse da zero

- Una SBA ha esattamente T componenti diverse da zero

Riconoscere le SBA in pratica

- Una SBA ha esattamente T componenti diverse da zero



In ogni periodo k deve essere: $x_k + s_{k-1} > 0$

In ognuno dei T periodi esattamente una delle variabili x_k, s_{k-1} è diversa da zero ($x_k \cdot s_{k-1} = 0$)

Se (x, s) è una SBA, in ogni periodo k abbiamo uno dei due casi:

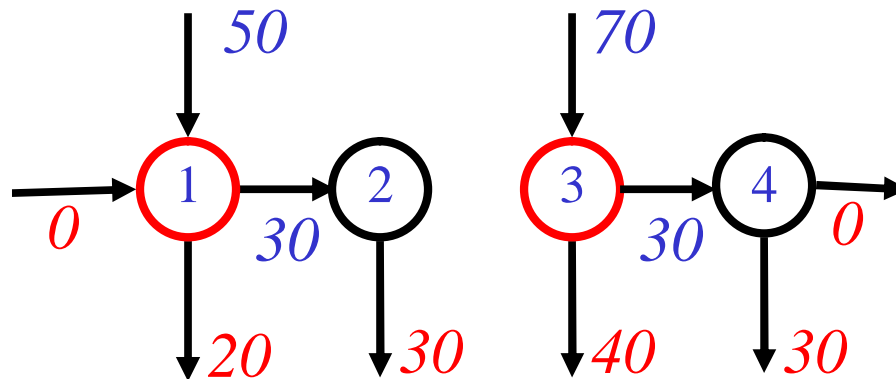
1. *Produzione positiva* ($x_k > 0$) e *Scorte nulle* ($s_{k-1} = 0$)
2. *Produzione nulla* ($x_k = 0$) e *Scorte positive* ($s_{k-1} > 0$)

Periodi Produttivi

Se (x, s) è una SBA, in ogni periodo k abbiamo uno dei due casi:

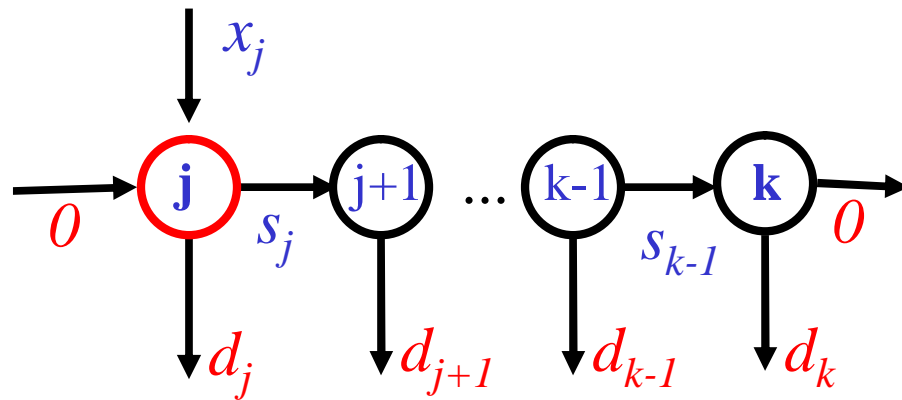
1. *Produzione positiva* ($x_k > 0$) e *Scorte nulle* ($s_{k-1} = 0$)
2. *Produzione nulla* ($x_k = 0$) e *Scorte positive* ($s_{k-1} > 0$)

- $k \in [1, \dots, T]$ è un periodo produttivo se $x_k > 0$
- La domanda di un periodo non produttivo h è soddisfatta dalla produzione nell'ultimo periodo produttivo $k < h$.



- L'insieme di periodi la cui domanda è soddisfatta da uno specifico periodo produttivo viene detto intervallo di produzione
- All'inizio e alla fine di un intervallo di produzione la *giacenza è nulla*

Intervallo di Produzione



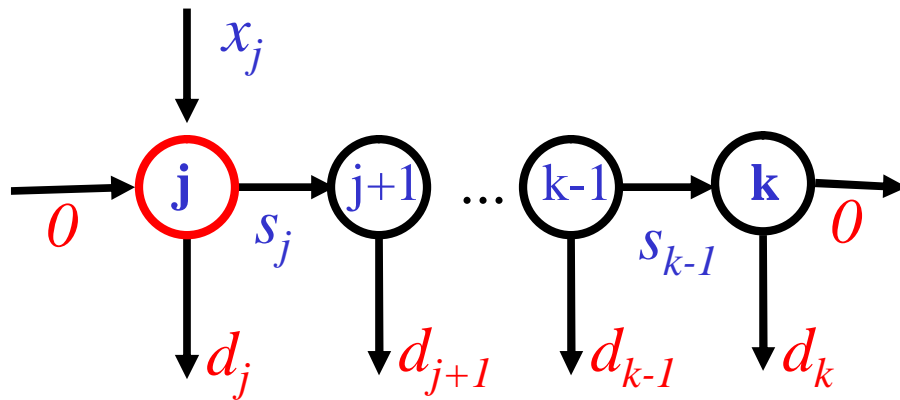
Intervallo di Produzione
[$j, j+1, j+2, \dots, k-1, k$]

Definiti gli estremi di un intervallo di produzione è possibile calcolare quantità prodotte e immagazzinate

$$\bar{x}_j = \sum_{r=j}^k d_r \quad \bar{x}_t = 0 \quad t = j+1, \dots, k$$

$$\bar{s}_t = \sum_{r=t+1}^k d_r \quad t = j, j+1, \dots, k-1 \quad \bar{s}_k = 0$$

Intervallo di Produzione (II)



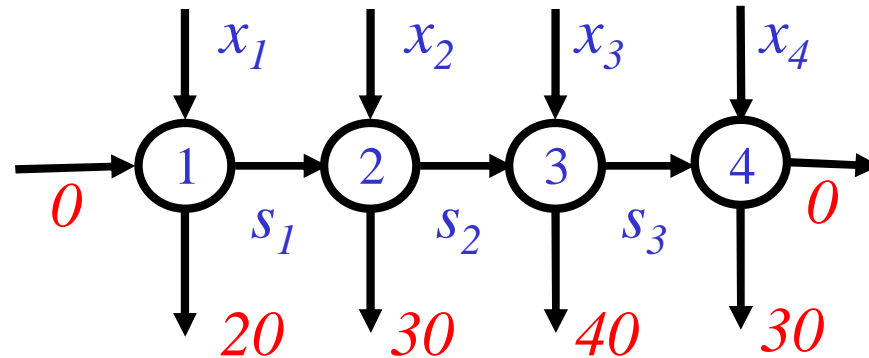
Intervallo di Produzione
 $[j, \dots, k] = [j, j+1, j+2, \dots, k-1, k]$

Definiti gli estremi di un intervallo di produzione è possibile calcolare il costo di produzione e stoccaggio per servire la domanda dei periodi contenuti in $[j, \dots, k]$

$$M(j, k) = \sum_{t=j}^k (C_t(\bar{x}_t) + H_t(\bar{s}_t))$$

Esempio Calcolo della Matrice M

t	d_j	A	c	h
1	20	30	3	2
2	30	40	3	2
3	40	30	4	1
4	30	50	4	-

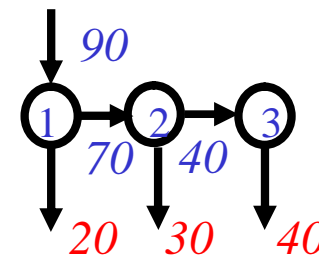


- Costo di Produzione Lineare (con costo fisso):

$$C_t(x_t) = A_t \eta(x_t) + c_t x_t$$

- Costo di Stoccaggio Lineare (costo fisso nullo)

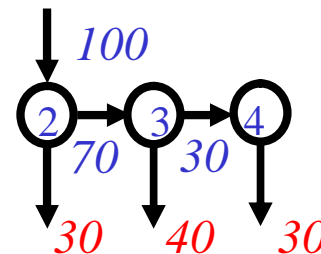
$$H_t(s_t) = h_t s_t$$



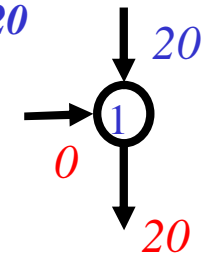
$$M(1,3) = 30 + 90 \cdot 3 + 70 \cdot 2 + 40 \cdot 2 = 520$$

$M =$

	1	2	3	4
1	90	240	520	760
2	-	130	330	510
3	-	-	190	340
4	-	-	-	170

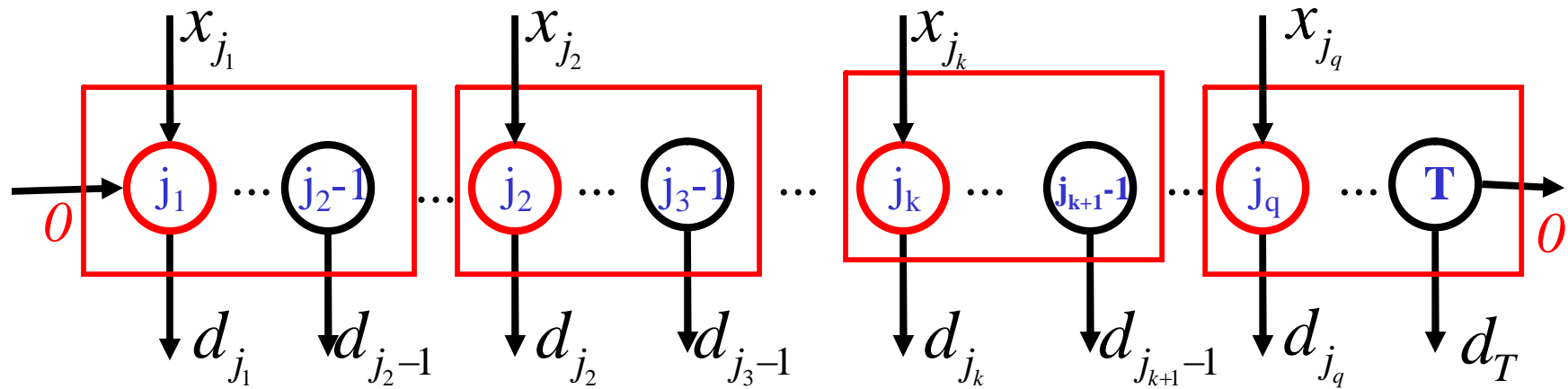


$$M(2,4) = 40 + 100 \cdot 3 + 70 \cdot 2 + 30 \cdot 1 = 510$$



$$M(1,1) = 30 + 20 \cdot 3 = 90$$

Struttura di una Soluzione di Base



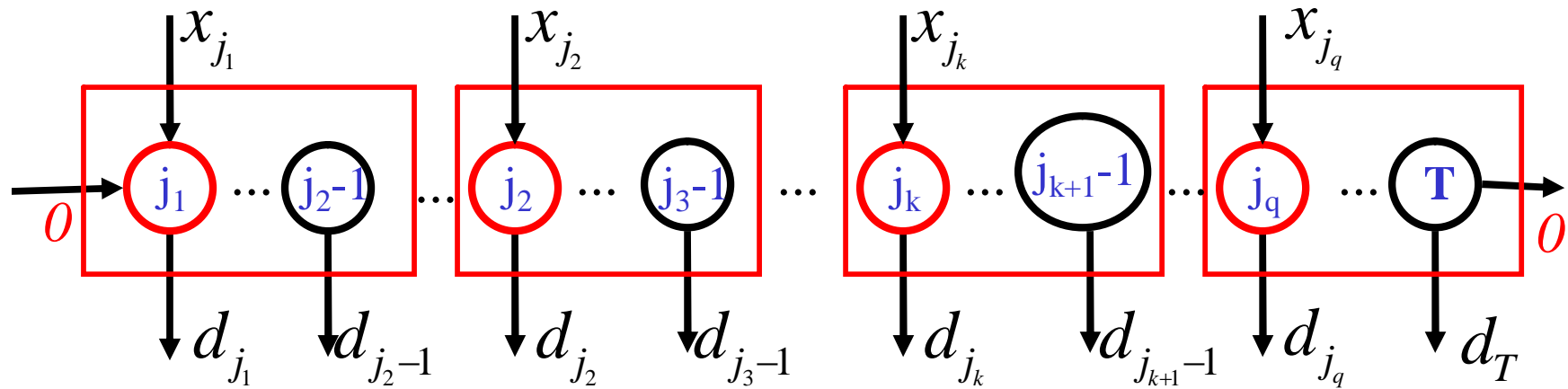
Insieme dei periodi produttivi di (x, s) :

$$J = \{ j_1, j_2, \dots, j_q \} \subseteq [1, \dots, T] \quad (x_1 > 0 \rightarrow j_1 = 1)$$

Caratterizza una SBA:

$$\begin{aligned} f(x, s) &= \sum_{t=1}^T (C_t(x_t) + H_t(s_t)) = \\ &= \sum_{r=1}^{q-1} M(j_r, j_{r+1} - 1) + M(j_q, T) = Z(J) \end{aligned}$$

Funzione Obiettivo (Funzione d'Insieme)



Problema di Programmazione della Produzione:

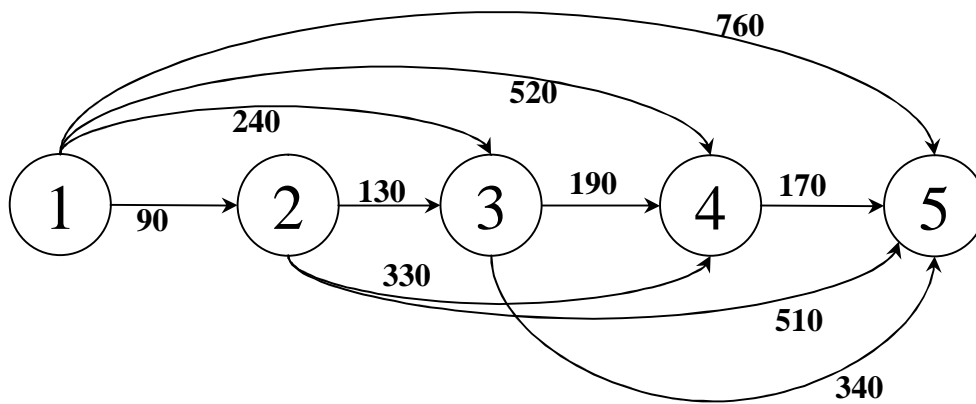
$$\min Z(J)$$

$$J \subseteq [1, \dots, T]; \quad J \neq \emptyset$$

TROVARE: J^* : $Z(J^*) \leq Z(J)$ per ogni $J \subseteq [1, \dots, T]$

Il grafo dinamico

- Serve a ridurre il problema in esame a un problema di cammino minimo su grafo orientato aciclico (*DAG*)
- Grafo orientato $G(V,A)$ con lunghezze degli archi $l \in R^A$
- $V = \{1, \dots, T+1\}$ (tanti nodi quanti periodi più un nodo ulteriore)
- $A = \{(i,j): i \in V, j \in V, i < j\}$ (tutti gli archi ‘*in avanti*’)
- $l_{ij} = M(i, j-1)$, per ogni $(i,j) \in A$.
- OBS: V ordinato topologicamente: per ogni $(u,v) \in A \rightarrow u < v$



$M =$

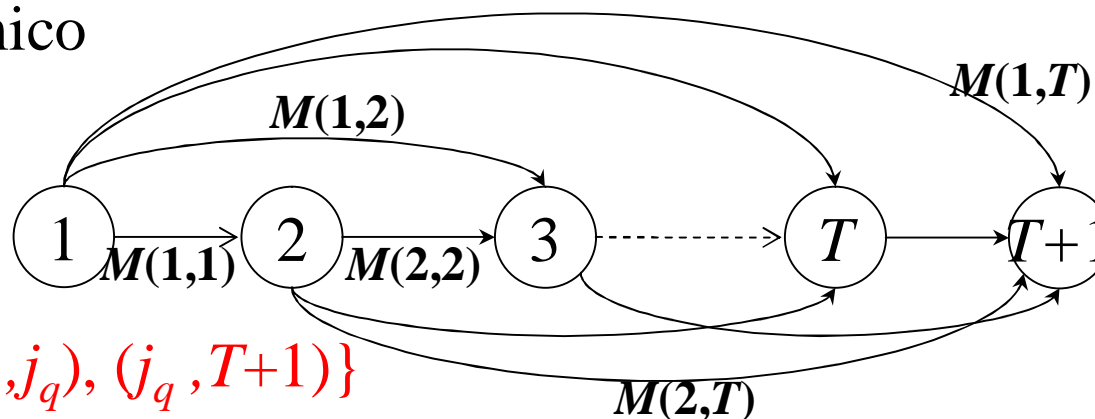
	1	2	3	4
1	90	240	520	760
2	-	130	330	510
3	-	-	190	340
4	-	-	-	170

Esempio: $T = 4$

Cammini e insiemi produttivi

- A ogni insieme di periodi produttivi $J = \{j_1, j_2, \dots, j_q\}$ ($j_1=1$) corrisponde biunivocamente un cammino orientato P dal nodo 1 al nodo $T+1$ nel grafo dinamico

$$J = \{j_1, j_2, j_3, \dots, j_q\}$$



$$P = \{(1, j_2), (j_2, j_3), \dots, (j_{q-1}, j_q), (j_q, T+1)\}$$

P passa esattamente per i nodi corrispondenti ai periodi di J e termina nel nodo $T+1$

- Il costo di J e la lunghezza di P coincidono. Infatti:

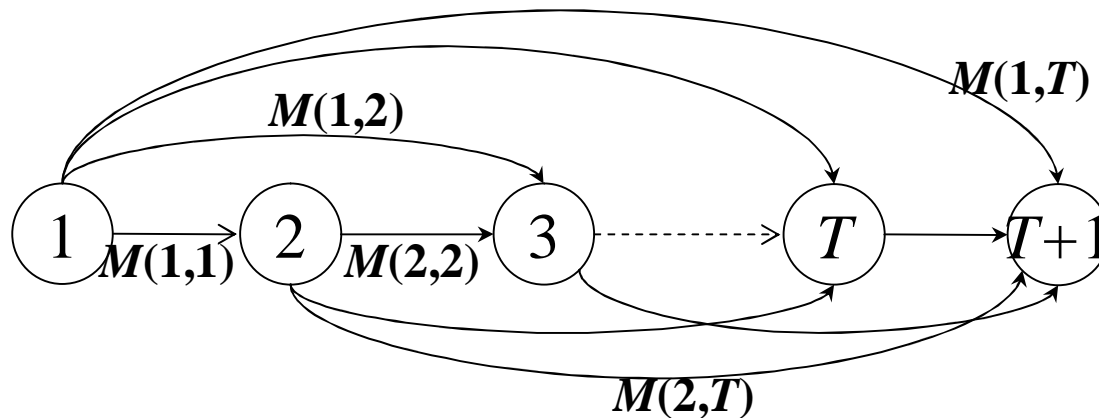
$$Z(J) = M(1, j_2-1) + M(j_2, j_3-1) + \dots + M(j_{q-1}, j_q-1) + M(j_q, T)$$

$$L(P) = l_{1j_2} + l_{j_2j_3} + \dots + l_{j_{q-1}j_q} + l_{j_qT+1} =$$

$$M(1, j_2-1) + M(j_2, j_3-1) + \dots + M(j_{q-1}, j_q-1) + M(j_q, T)$$

Cammini minimi in grafi aciclici

- La lunghezza del cammino di lunghezza minima corrisponde quindi al costo dell'insieme produttivo di costo minimo
- Il problema originale può essere risolto utilizzando un algoritmo per il calcolo del cammino minimo su grafo aciclico.

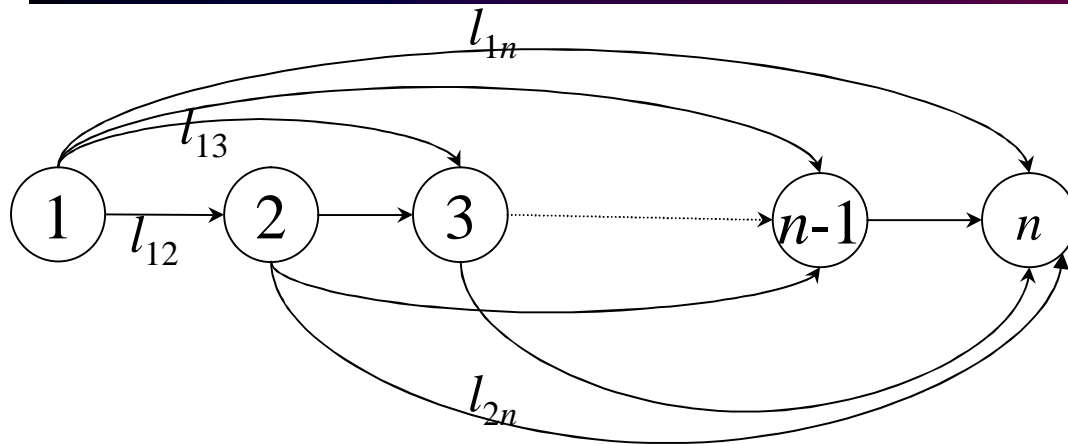


Sia F_k = *lunghezza cammino minimo da 1 a k+1*

F_T costo minimo di produzione e stoccaggio per soddisfare la domanda sull'orizzonte temporale

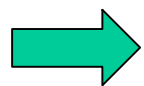
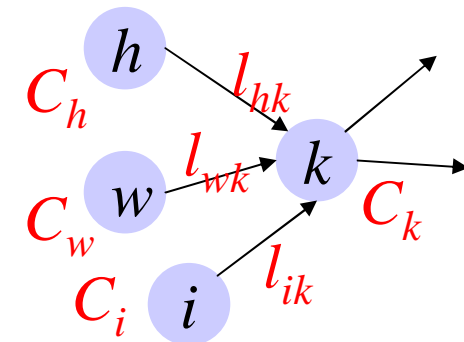
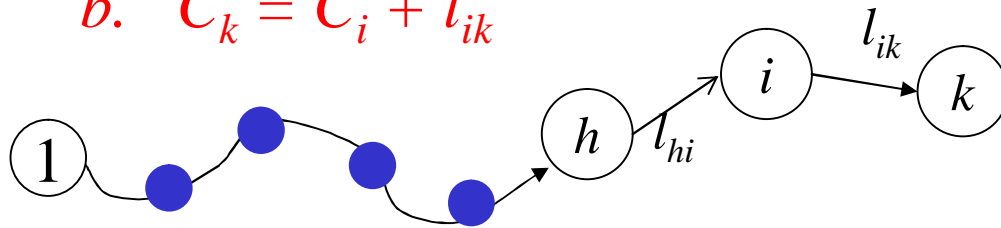
Nota: F_k costo minimo di produzione e stoccaggio per soddisfare la domanda fino a k .

Calcolo cammino minimo in grafi aciclici



$G = (V,A)$ grafo aciclico
 V ordinato topologicamente
 l_{ik} peso arco ik
 $C_k =$ lunghezza cammino minimo da 1 a k

- Ogni cammino da 1 a k può passare solo per nodi $i < k$.
- Se $P = \{(1, v_2), \dots, (h, i), (i, k)\}$ cammino minimo da 1 a k allora:
 - $P' = \{(1, v_2), \dots, i\}$ cammino minimo da 1 a i
 - $C_k = C_i + l_{ik}$

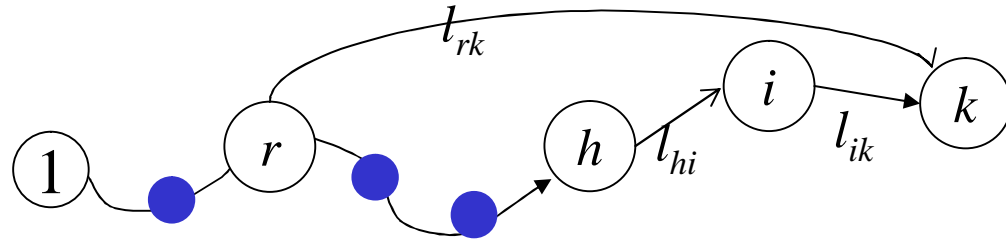


Se conosciamo C_j per ogni $j < k$, possiamo calcolare C_k

$$C_k = \min_{j \in \{1, \dots, k-1\}} \{C_j + l_{jk}\}$$

Algoritmo ricorsivo calcolo cammino minimo

- Se $r \in V$ è tale che $C_k = C_r + l_{rk} = \min_{j \in \{1, \dots, k-1\}} \{C_j + l_{jk}\}$



r è il predecessore $p(k)$ di k sul cammino minimo da 1 a k

Calcolo del cammino minimo in grafo aciclico $G(V,A)$, $l \in R^A$

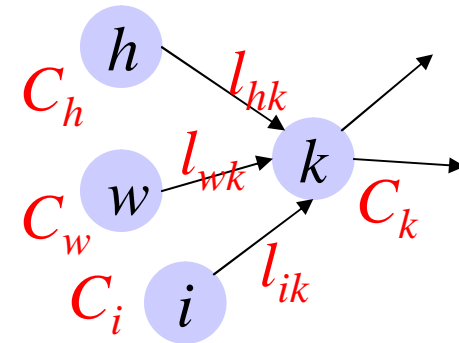
Assunzione: nodi di ordinati topologicamente

Inizializzazione: $C_1 = 0$, $p(1) = *$

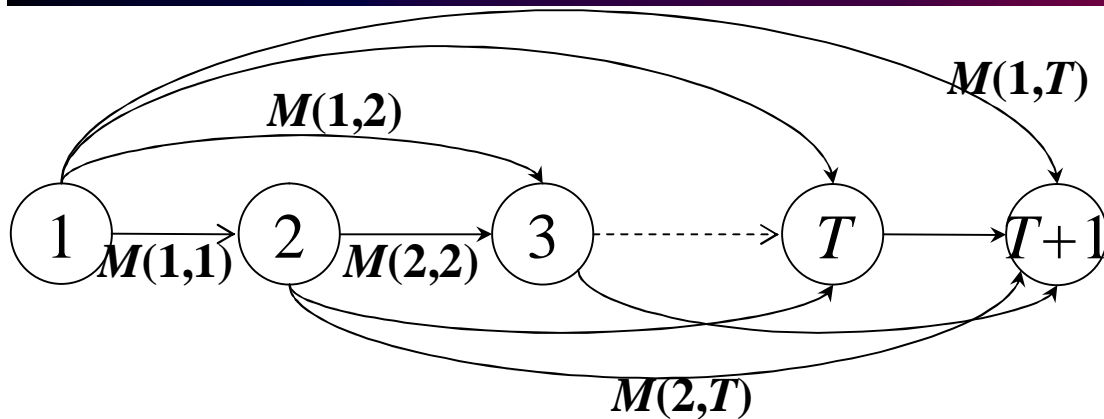
for $j = 2$ to n

$$C_k = \min_{j < k} \{C_j + l_{jk}\}$$

$$p(k) = \operatorname{argmin}_{j < k} \{C_j + l_{jk}\}$$



Cammino minimo sul grafo dinamico



$G(V,A) \quad l \in \mathbb{R}^A$

$$l_{jk+1} = M(j,k)$$

- Ricordando che, nel grafo dinamico, F_k indica la *lunghezza cammino minimo da 1 a $k+1$* $\rightarrow F_k = C_{k+1}$

$$C_{k+1} = \min_{j \in \{1, \dots, k\}} \{C_j + l_{jk+1}\} \quad k = 1, \dots, T$$

$$\text{con } C_1 = 0$$

$$\rightarrow F_k = \min_{j \in \{1, \dots, k\}} \{F_{j-1} + M(j, k)\} \quad k = 1, \dots, T$$

$$\text{con } F_0 = 0$$

Sia $L(k)$ il periodo produttivo che serve il periodo k

$$\rightarrow L(k) = \arg \min_{j \in \{1, \dots, k\}} \{F_{j-1} + M(j, k)\} \quad k = 1, \dots, T$$

Algoritmo di Wagner-Whitin (idea)

$$F_k = \min_{j \in \{1, \dots, k\}} \{ F_{j-1} + M(j, k) \}$$

$$F_0 := 0$$

$$F_1 := \min \{ F_0 + M(1,1) \}$$

$$F_2 := \min \{ F_0 + M(1,2), F_1 + M(2,2) \}$$

.....

F_T

valore della soluzione ottima

Algoritmo di Wagner-Whitin

FASE I: Inizializzazione (calcolo matrice M)

for $k:=1$ *to* T

for $j:=1$ *to* k

$$\bar{x}_j = \sum_{r=j}^k d_r$$

$$\bar{s}_t = \sum_{r=t+1}^k d_r \quad t = j, j+1, \dots, k-1$$

$$M(j, k) = C_j(\bar{x}_j) + \sum_{t=j}^{k-1} H_t(\bar{s}_t)$$

end for

Algoritmo di Wagner-Whitin

FASE I: Calcolo del valore ottimo

$$F_0 := 0$$

for $k:=1$ to T

$$F_k = \min_{j \in \{1, \dots, k\}} \{ F_{j-1} + M(j, k) \}$$

$$L(k) = \arg \min_{j \in \{1, \dots, k\}} \{ F_{j-1} + M(j, k) \}$$

end for

$L(k)$ periodo produttivo che serve il periodo k

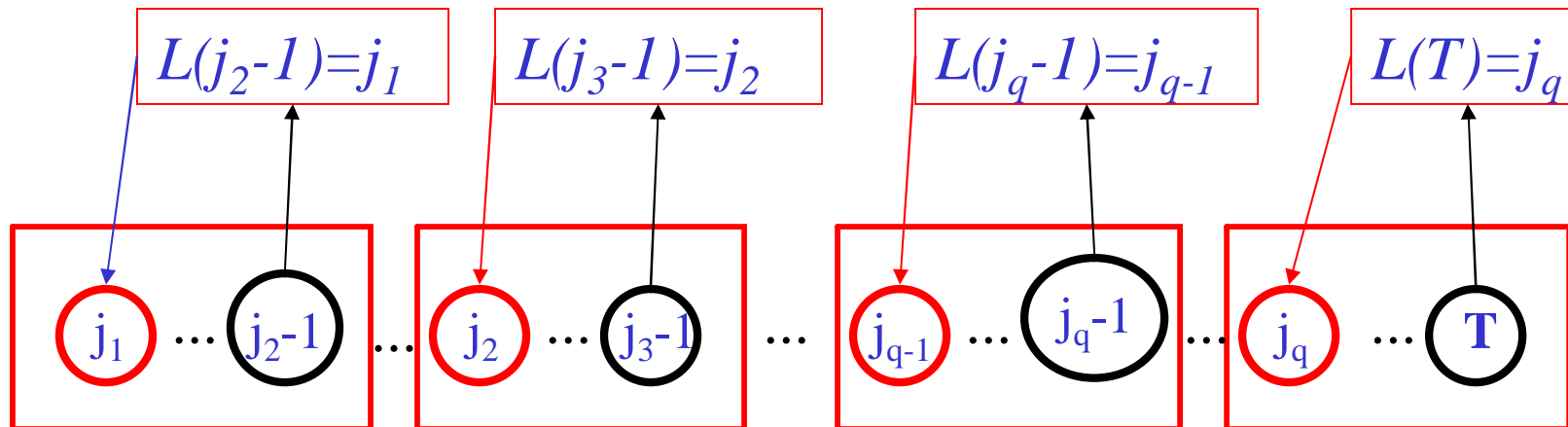
F_T

valore della soluzione ottima

Algoritmo di Wagner-Whitin

FASE II: Individuazione della Soluzione J^*

$L(k)$ periodo produttivo che serve il periodo k



Algoritmo di Wagner-Whitin

FASE II: Individuazione della Soluzione J^*

$$J^* := \{L(T)\}$$

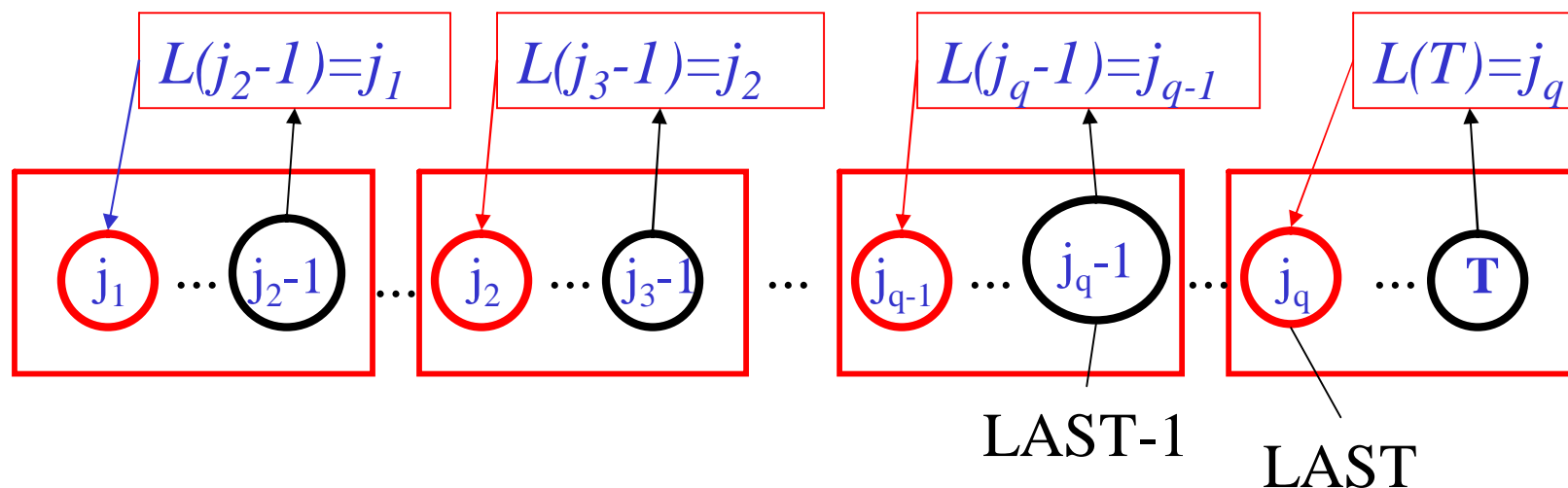
$$LAST := L(T)$$

while $LAST \neq 1$ *do*

$$LAST := L(LAST - 1)$$

$$J^* := J^* \cup \{LAST\}$$

endwhile



Esempio di Programmazione della Produzione

- Modello Deterministico
- Controllo Discreto (quattro periodi) [1,2,3,4]
- Singlo bene

- Domanda *Variabile* nel tempo

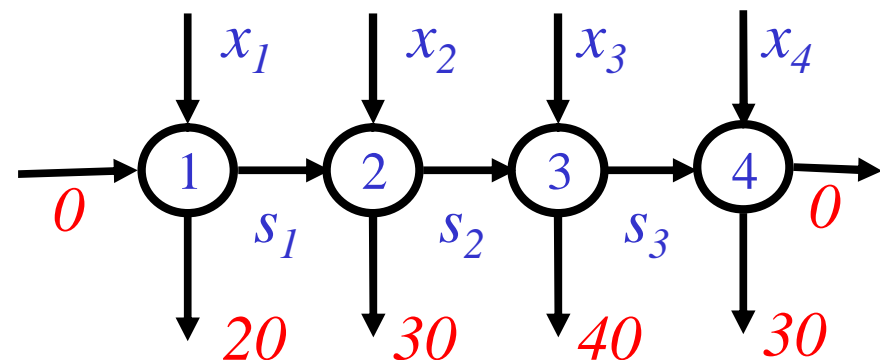
- Costo di Produzione Lineare (con costo fisso):

$$C_t(x_t) = A_t \eta(x_t) + c_t x_t$$

- Costo di Stoccaggio Lineare (costo fisso nullo)

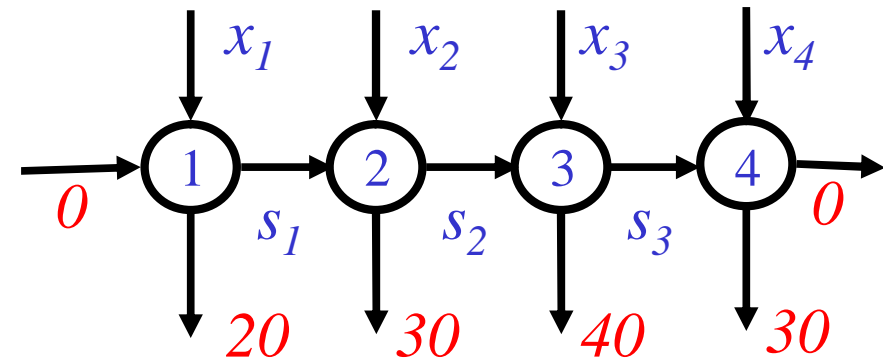
$$H_t(s_t) = h_t s_t$$

Periodi	Domanda	A	c	h
1	20	30	3	2
2	30	40	3	2
3	40	30	4	1
4	30	50	4	-



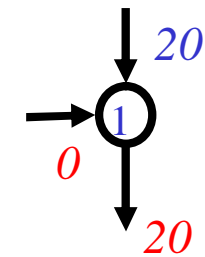
Calcolo della Matrice M

Periodi	Domanda	A	c	h
1	20	30	3	2
2	30	40	3	2
3	40	30	4	1
4	30	50	4	-

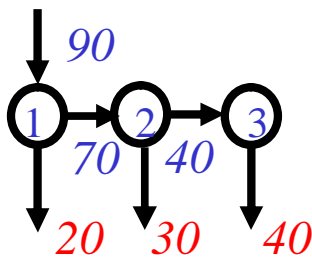


$M =$

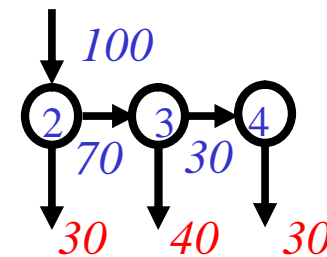
	1	2	3	4
1	90	240	520	760
2	-	130	330	510
3	-	-	190	340
4	-	-	-	170



$$M(1,1) = 30 + 20 * 3 = 90$$



$$M(1,3) = 30 + 90 * 3 + 70 * 2 + 40 * 2 = 520$$



$$M(2,4) = 40 + 100 * 3 + 70 * 2 + 30 * 1 = 510$$

Algoritmo di Wagner-Whitin: Fase 1

$$F_k = \min_{j \in \{1, \dots, k\}} \{ F_{j-1} + M(j, k) \}$$

$$F_0 := 0$$

$$F_1 := \min\{ F_0 + M(1,1) \} = 90$$

$$\begin{aligned} F_2 &:= \min\{ F_0 + M(1,2), F_1 + M(2,2) \} = \\ &= \min\{ 0 + 240, 90 + 130 \} = 220 \quad (L_2 = 2) \end{aligned}$$

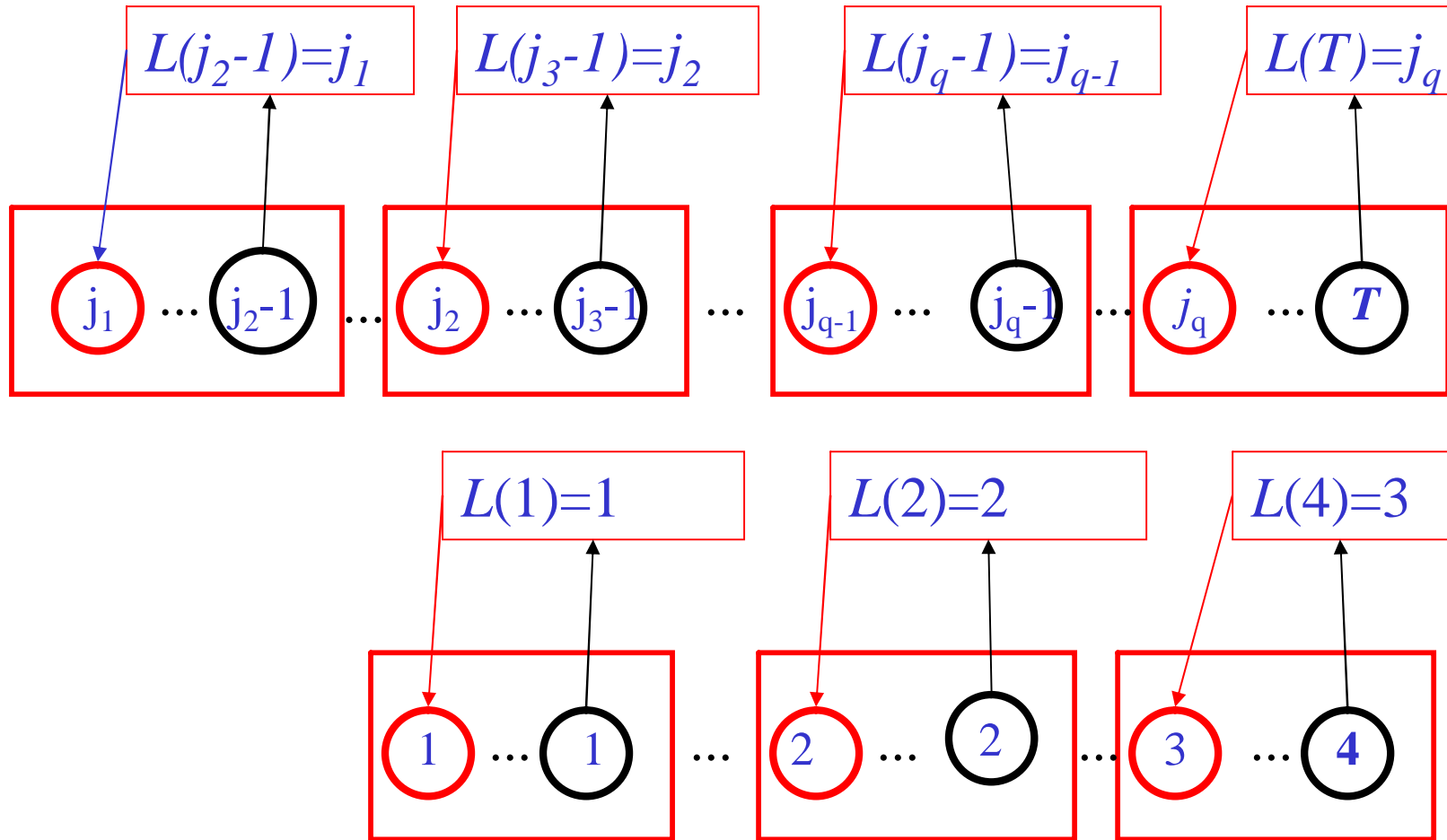
$$\begin{aligned} F_3 &:= \min\{ F_0 + M(1,3), F_1 + M(2,3), F_2 + M(3,3) \} = \\ &= \min\{ 0 + 520, 90 + 330, 220 + 190 \} = 410 \quad (L_3 = 3) \end{aligned}$$

$$\begin{aligned} F_4 &:= \min\{ F_0 + M(1,4), F_1 + M(2,4), F_2 + M(3,4), F_3 + M(4,4) \} = \\ &= \min\{ 0 + 760, 90 + 510, 220 + 340, 410 + 170 \} = 560 \quad (L_4 = 3) \end{aligned}$$

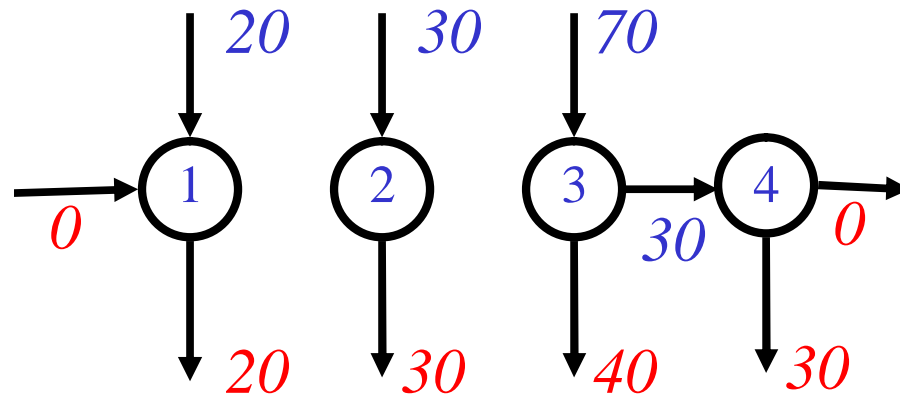
Algoritmo di Wagner-Whitin: Fase 2

FASE II: Individuazione della Soluzione J^*

L_k periodo produttivo che serve il periodo k



Soluzione Ottima Trovata



Controllo discreto: formulazione

Formulazione lineare con costi fissi

- In presenza di vincoli aggiuntivi, l'algoritmo di Wagner e Within non può essere applicato
- Una formulazione matematica può in genere essere facilmente emendata per tenere conto di ulteriori vincoli.
- Consideriamo il caso di costi fissi di produzione e costi variabili che variano linearmente con la produzione e le scorte:

Costo di Produzione nel periodo t

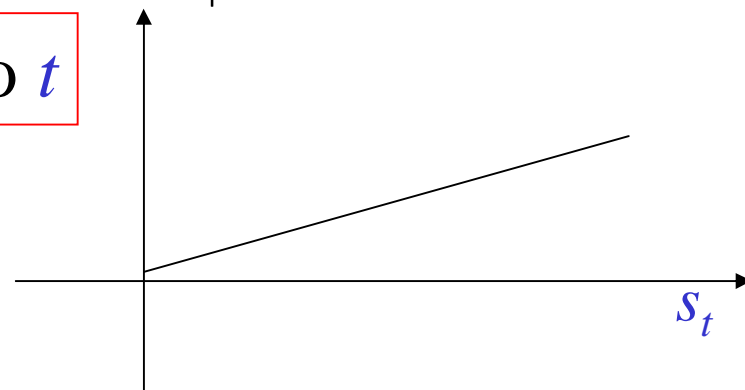
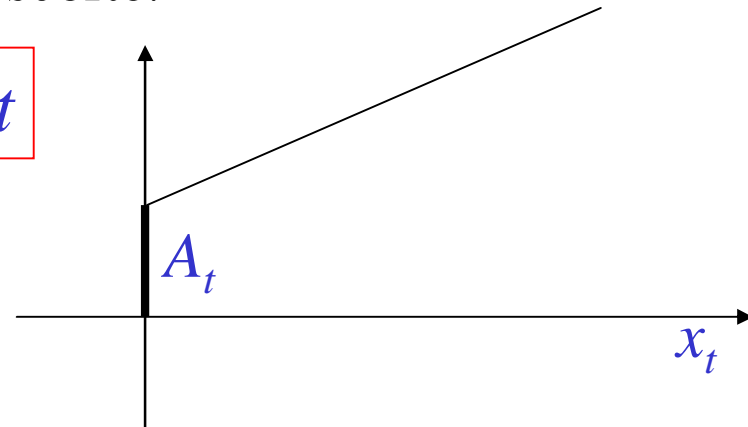
Costo fisso + costo lineare

$$C_t(x_t) = A_t \eta(x_t) + c_t \cdot x_t$$

Costo di Stoccaggio nel periodo t

costo lineare

$$H_t(s_t) = h_t \cdot s_t$$



Modellazione costi fissi

- Introduciamo una variabile binaria z_k per ogni periodo k

$$z_k = \begin{cases} 0 & \text{se } x_k = 0 \\ 1 & \text{se } x_k > 0 \end{cases}$$

- Per semplificare la notazione reintroduciamo le variabili s_0 e s_T

$$\min \sum_{k=1}^T c_k x_k + \sum_{k=1}^T A_k z_k + \sum_{k=1}^T h_k s_k$$

$$x_k + s_{k-1} - s_k = d_k \quad k = 1, \dots, T$$

$$x_k \leq R_k \cdot z_k \quad k = 1, \dots, T \quad \text{Vincoli di } upper \text{ bound variable}$$

$$x_k, s_k \geq 0 \quad k = 1, \dots, T$$

$$s_0, s_T = 0$$

- R_k costante “sufficientemente” grande

Ricordando che
$$x_k \leq \sum_{j=k}^T d_j \quad k = 1, \dots, T$$

Poniamo
$$R_k = \sum_{j=k}^T d_j \quad k = 1, \dots, T$$

Cammini minimi su grafi aciclici

Algoritmo per i Cammini Minimi

- L'algoritmo di programmazione dinamica è un esempio di calcolo di cammino minimo su grafi orientati privi di cicli orientati.
- Lo stesso algoritmo (*mutatis mutandis*) può essere applicato a grafi orientati aciclici generici (DAG = directed acyclic graph)
- Problema 1: come stabilire se un grafo orientato è aciclico?
- Problema 2: Come costruire un ordinamento topologico dei nodi (se esiste)?

I due problemi in realtà sono due facce della stessa medaglia.

Teorema della numerazione topologica

Def. *Numerazione topologica* dei nodi di un grafo orientato $G(V,A)$: numerazione dei nodi $V = \{1,2, \dots, n\}$ tale che $(k,i) \in A \rightarrow k < i$.

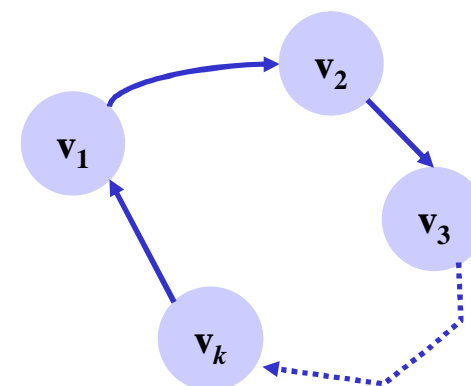
Teorema 3.8 (*della numerazione topologica*): un grafo orientato $G(V,A)$ ammette una numerazione topologica dei nodi se e solo se G non contiene cicli orientati (= G è un *DAG*)

DIMOSTRAZIONE: Solo Se (Necessità). Per assurdo.

Supponiamo i nodi di G numerati topologicamente e che G contenga almeno un ciclo orientato $\{(v_1v_2), \dots, (v_{k-1}v_k), (v_kv_1)\}$

Essendo $v_1v_2 \in A$ deve essere $v_1 < v_2$

Similmente $v_2 < v_3 < \dots < v_k < v_1$ e quindi $v_1 < v_1$, contraddizione



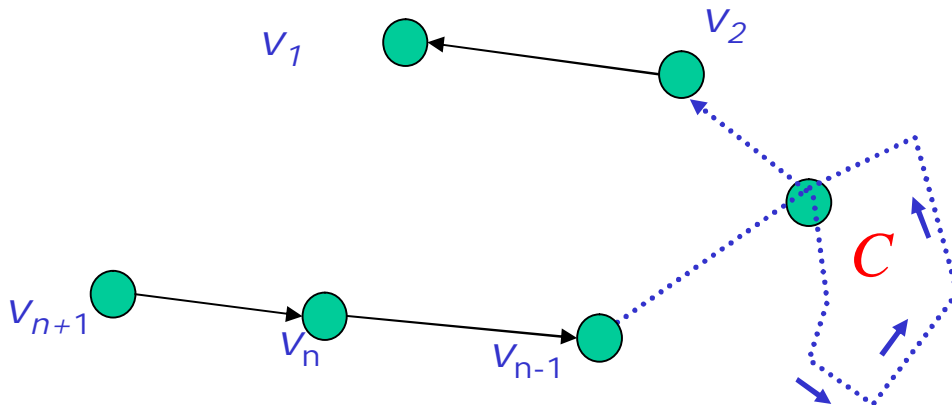
Dimostrazione

Sufficienza (divisa in due parti). **Parte 1:**

Se $G(V,A)$ è un DAG allora esiste un nodo s privo di archi entranti (*sorgente*)

Per assurdo: supponiamo che per ogni $v \in V$ esista $uv \in A$. Sia $|V| = n$.
Scegliamo un nodo iniziale v_1 . Per ipotesi esiste $(v_2, v_1) \in A$.

Ma esiste anche $(v_3, v_2) \in A$. Ripetendo n volte, identifichiamo un walk orientato $W = \{(v_{n+1}v_n), (v_nv_{n-1}), \dots, (v_2v_1)\}$ che comprende $n+1$ nodi. Ma $|V| = n$ quindi almeno un nodo appare due volte in W



Quindi W contiene un cammino chiuso orientato C



Algoritmo per la numerazione topologica

Parte 2. La dimostrazione è costruttiva.

- Poni $G_1 = G$. Scegli una sorgente s_1 di G_1 .
- Costruisci $G_2 = G_1[V - \{s_1\}]$ indotto in G_1 dai nodi $V - \{s_1\}$ (i.e. ottenuto da G rimuovendo il nodo s_1 e gli archi incidenti in s_1)
- G_2 è aciclico e quindi contiene almeno una sorgente s_2
- L'argomento può essere riapplicato a G_2 rimuovendo s_2 e costruendo il sottografo G_3 e così di seguito
- Si ottiene la sequenza di grafi (G_1, G_2, \dots, G_n) e di nodi distinti (s_1, s_2, \dots, s_n) con s_i sorgente nel grafo G_i , per $i = 1, \dots, n$
- La numerazione $(s_1 = 1, s_2 = 2, \dots, s_n = n)$ è topologica, infatti:
 1. Per $i = 1, \dots, n$, i nodi di G_i sono $V_i = \{i, i+1, \dots, n\}$.
 2. Il nodo i è sorgente in $G_i \rightarrow$ non esiste $k \in \{i+1, \dots, n\}$ con $ki \in A$ (altrimenti $ki \in G_i$ e i non è una sorgente di G_i)



Calcolo dell'ordinamento topologico

L'algoritmo è ispirato alla dimostrazione (costruttiva) di esistenza della numerazione topologica:

Algoritmo generico di numerazione topologica

Inizializzazione: *Poni* $G_1 = G$

for $j = 1$ to n

Trova una sorgente s_j in G_j

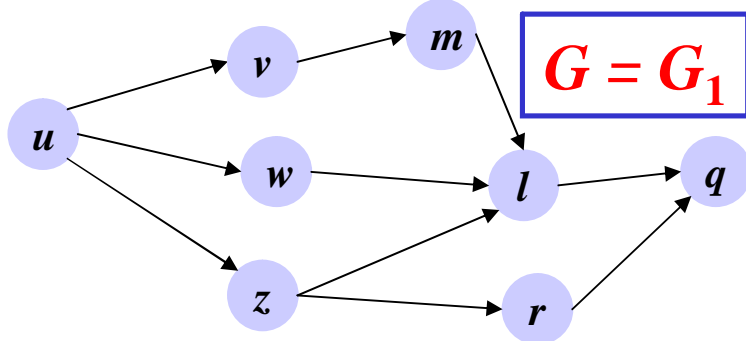
Assegna a s_j indice j

Poni $G_{j+1} = G_j - \{s_j\}$ (rimuovi s_j e tutti gli archi incidenti)

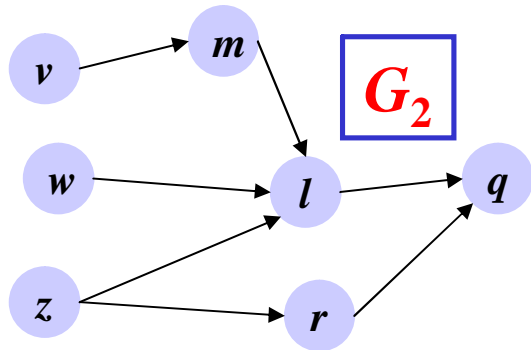
Endfor

Ordinare un grafo topologicamente può essere svolto in $O(|A|)$

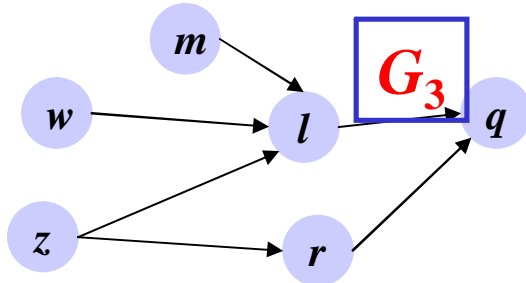
Esempio di numerazione topologica



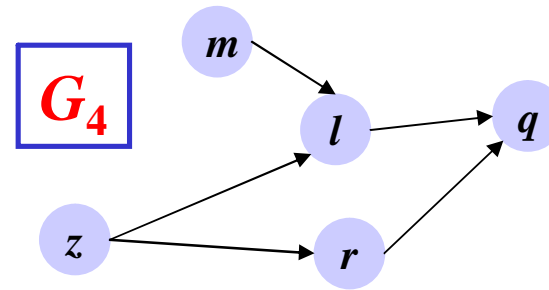
u è sorgente in G_1 . Poni $u = 1$, rimuovilo costruendo G_2



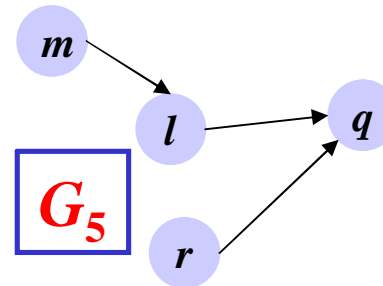
v è sorgente in G_2 . $v = 2$



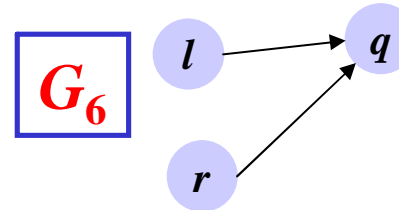
w è sorgente in G_3 . $w = 3$



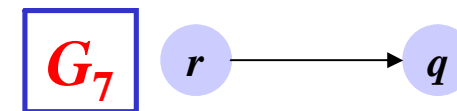
z è sorgente in G_4 . $z = 4$



m è sorgente in G_5 . $m = 5$



l è sorgente in G_6 . $l = 6$

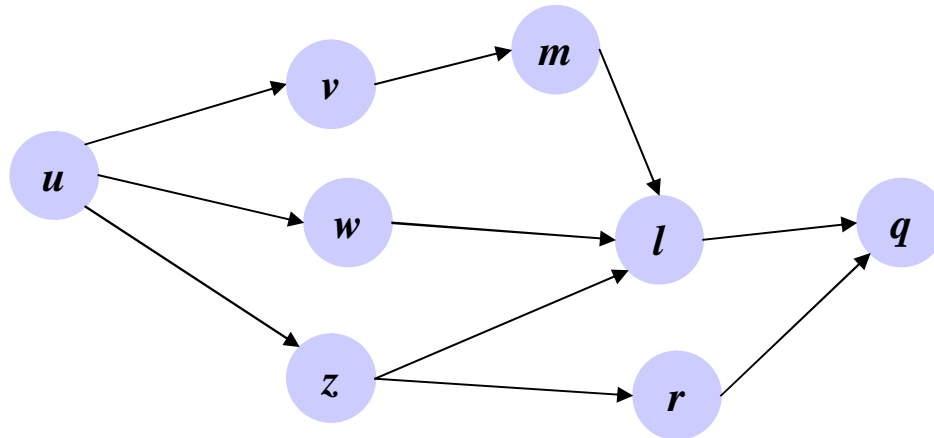


r è sorgente in G_7 . $r = 7$

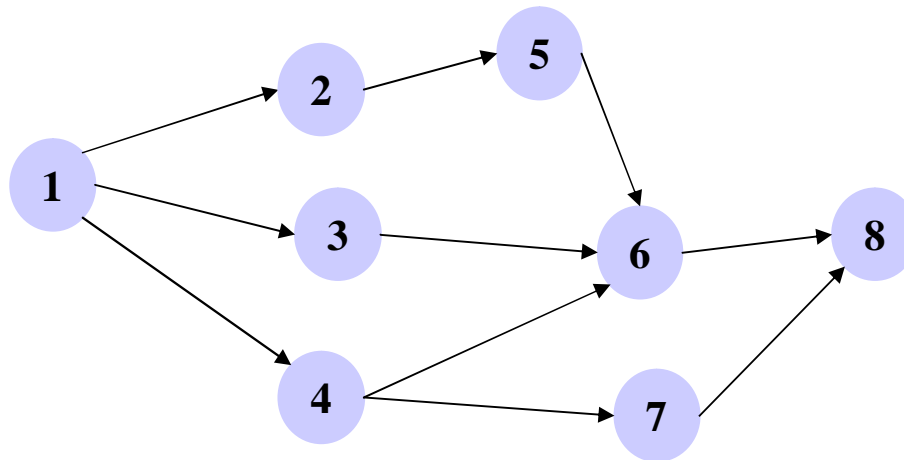


q è sorgente in G_8 . $q = 8$

Esempio di numerazione topologica



Numerando i nodi nell'ordine in cui li abbiamo scelti e rimossi otteniamo la numerazione topologica:



Calcolo del cammino minimo per DAG

Calcolo del cammino minimo grafi aciclici

Ordina topologicamente i nodi di $G = (V, A)$

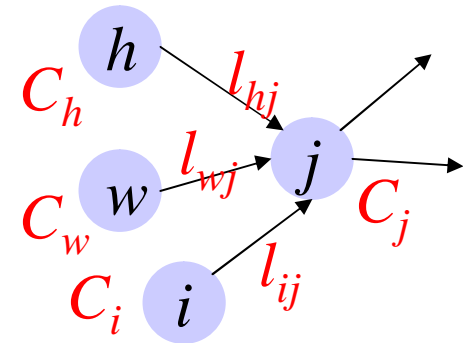
Inizializzazione: $C_1 = 0, p_1 = *$.

for $j = 2$ to n

$$C_j = \min\{C_i + l_{ij} \mid i \in N^-(j)\}$$

$$p_j = \operatorname{argmin}\{C_i + l_{ij} \mid i \in N^-(j)\}$$

Endfor

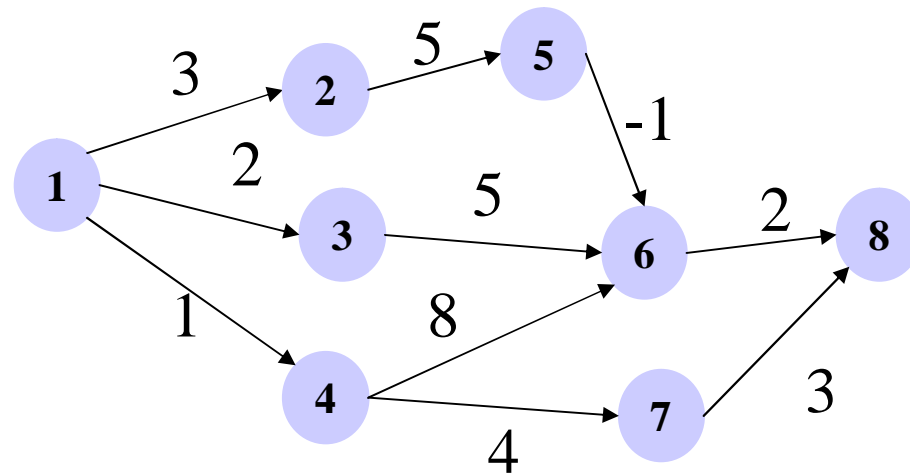


- A ogni iterazione visito la stella entrante nel nodo j
- Alla fine avrò visitato tutti gli archi esattamente una volta.

Complessità : $O(m)$

- Poiché anche la numerazione topologica ha complessità $O(m)$, complessivamente la complessità resta $O(m)$

Esempio di calcolo del cammino minimo



$$C_1 = 0$$

$$C_2 = \min \{ C_1 + l_{12} \} = \min \{ 3 \} = 3$$

$$C_3 = \min \{ C_1 + l_{13} \} = \min \{ 2 \} = 2$$

$$C_4 = \min \{ C_1 + l_{14} \} = \min \{ 1 \} = 1$$

$$C_5 = \min \{ C_2 + l_{25} \} = \min \{ 8 \} = 8$$

$$C_6 = \min \{ C_3 + l_{36}, C_4 + l_{46}, C_5 + l_{56} \} = \min \{ 7, 9, 7 \} = 7$$

$$C_7 = \min \{ C_4 + l_{47} \} = \min \{ 5 \} = 5$$

$$C_8 = \min \{ C_6 + l_{68}, C_7 + l_{78} \} = \min \{ 9, 8 \} = 8$$

$$p_1 = *$$

$$p_2 = 1$$

$$p_3 = 1$$

$$p_4 = 1$$

$$p_5 = 2$$

$$p_6 = 5$$

$$p_7 = 4$$

$$p_8 = 7$$

Altri esempi di programmazione dinamica

Programmazione dinamica e knapsack 0,1

Problema del knapsack binario:

Dati n oggetti $\{1, \dots, n\}$, di valore $c_j \geq 0$ e ingombro $b_j \geq 0$, con $j = 1, \dots, n$ e dato uno zaino di capacità K

- trovare il sottoinsieme di oggetti J il cui volume complessivo non ecceda la capacità dello zaino e che abbia valore massimo

Siano $c^j = (c_1, \dots, c_j)$, $b^j = (b_1, \dots, b_j)$ i vettori di valore e ingombro associati ai primi j oggetti. Il problema di knapsack è completamente definito dalla tripla (c^n, b^n, K)

Per ogni $j \in \{1, \dots, n\}$, introduciamo una variabile binaria x_j

$$x_j = \begin{cases} 1 & \text{se l'oggetto } j \text{ è preso} \\ 0 & \text{altrimenti} \end{cases}$$

Formulazione:

$$\max c_1 x_1 + \dots + c_n x_n$$

$$b_1 x_1 + \dots + b_n x_n \leq K$$

$$x \in \{0,1\}^n, \quad c, b \in \mathbb{Z}_+^n, \quad K \in \mathbb{Z}_+$$

Sequenze di sottoproblemi di knapsack

Dato un problema di knapsack su n oggetti (c^n, b^n, K)

- Associamo $(n+1) \times (K+1)$ sottoproblemi (c^j, b^j, i) , per $j = 0, \dots, n$ e $i = 0, \dots, K$, ove il problema (c^j, b^j, i) è ottenuto considerando solo i primi j oggetti e capacità dello zaino pari a i

$$M(i, j) = \max c_1 x_1 + \dots + c_j x_j$$

Formulazione:

$$b_1 x_1 + \dots + b_j x_j \leq i$$

$$x \in \{0,1\}^j$$

Per $i = 0, \dots, K, j = 0, \dots, n$

$M(i, j)$ = valore soluzione ottima del sottoproblema

$J(i, j)$ = insieme degli oggetti nella soluzione ottima del sottoproblema corrispondente.

OSS: $M(K, n)$ valore soluzione ottima problema iniziale

$J(K, n)$ soluzione ottima problema iniziale

Calcolo di $M(i,j)$

Quanto vale $M(i, j)$?

- Se l'oggetto $j \notin J(i,j)$ (j non è preso nella soluzione ottima), allora la soluzione ottima è uguale a quella ottenuta trascurando l'oggetto

➡ $M(i, j) = M(i, j-1) \quad J(i, j) = J(i, j-1)$

- Se l'oggetto $j \in J(i,j)$ (i.e. j è preso nella soluzione ottima), allora

1) j contribuisce al valore della soluzione per il suo valore c_j

2) j “consuma” una parte pari a b_j del volume i

3) $J' = J(i,j) - \{j\} \subseteq \{1, \dots, j-1\}$, i.e. J' contiene solo oggetti $< j$

- Principio di ottimalità: J' è la soluzione ottima per il problema con gli oggetti $\{1, \dots, j-1\}$ e volume $i - b_j$ (dimostrare)

➡ $J' = J(i - b_j, j - 1) \quad c(J') = M(i - b_j, j - 1)$

$$c(J) = M(i,j) = M(i - b_j, j - 1) + c_j$$

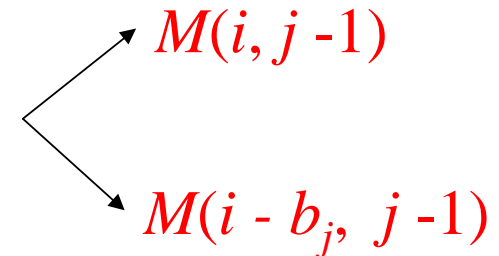
Formula Ricorsiva

A priori non so se l'oggetto j è contenuto nella soluzione $J(i,j)$,
ma lo posso valutare conoscendo $M(i, j - 1)$ e $M(i - b_j, j - 1)$

Infatti: $M(i, j) = \max \{ M(i - b_j, j - 1) + c_j, M(i, j - 1) \}$

Questa formula permette di calcolare ricorsivamente tutti i
valori della matrice M .

Posso quindi calcolare $M(i,j)$ se conosco



Ho bisogno di inizializzare la matrice

$$M(i, j) = \begin{cases} 0 & \text{se } i = 0 \text{ oppure } i > 0 \text{ e } j = 0 \\ -\infty & \text{se } i < 0 \end{cases}$$

L'algoritmo ricorsivo

Inizializza M

For $i = 1, \dots, K$

For $j = 1, \dots, n$

If $M(i - b_j, j-1) + c_j > M(i, j-1)$

then $M(i, j) = M(i - b_j, j-1) + c_j$

$J(i, j) = J(i - b_j, j-1) \cup \{j\}$

else $M(i, j) = M(i, j-1)$

$J(i, j) = J(i, j-1)$

EndIf

EndFor

EndFor

Complessità $O(nK)$ (pseudo-polinomiale)

Esempio

<i>i</i>		0	1	2	3	4	<i>j</i>
-	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	1	1	1	1	1	
3	0	1	2	2	2	2	
4	0						
5	0						
6	0						

$$\begin{aligned} \max x_1 + 2x_2 + x_3 + 3x_4 \\ 2x_1 + 3x_2 + 3x_3 + 4x_4 \leq 6 \\ x \in \{0,1\}^4 \end{aligned}$$

$$M(i, j) = \max \{M(i - b_j, j-1) + c_j, M(i, j-1)\}$$

$$M(1,1) = \max \{M(-1, 0)+1, M(1, 0)\} = 0$$

$$M(1,2) = M(1,3) = M(1,4) = 0$$

$$M(2,1) = \max \{M(0, 0)+1(=c_1), M(2, 0)\} = 1$$

$$M(2,2) = \max \{M(-1, 1)+2(=c_2), M(2, 1)\} = 1$$

$$M(2,3) = M(2,4) = 1$$

$$M(3,1) = \max \{M(1, 0)+1(=c_1), M(3, 0)\} = 1$$

$$M(3,2) = \max \{M(0, 1)+2(=c_2), M(3, 1)\} = 2$$

$$M(3,3) = \max \{M(0, 2)+1(=c_3), M(3, 2)\} = 2$$

$$M(3,4) = \max \{M(-1, 1)+3(=c_4), M(3, 3)\} = 2$$

Esempio

<i>i</i>		0	1	2	3	4	<i>j</i>
-	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	1	1	1	1	1	
3	0	1	2	2	2	2	
4	0	1	2	2	3	3	
5	0	1	3	3	3	3	
6	0						

$$\begin{aligned} \max x_1 + 2x_2 + x_3 + 3x_4 \\ 2x_1 + 3x_2 + 3x_3 + 4x_4 \leq 6 \\ x \in \{0,1\}^4 \end{aligned}$$

$$M(i, j) = \max \{M(i - b_j, j-1) + c_j, M(i, j-1)\}$$

$$M(4,1) = \max \{M(2, 0)+1, M(4, 0)\} = 1$$

$$M(4,2) = \max \{M(1, 1)+2, M(4, 1)\} = 2$$

$$M(4,3) = \max \{M(1, 2)+1, M(4, 2)\} = 2$$

$$M(4,4) = \max \{M(0, 3)+3, M(4, 3)\} = 3$$

$$M(5,1) = \max \{M(3, 0)+1, M(5, 0)\} = 1$$

$$M(5,2) = \max \{M(2, 1)+2, M(5, 1)\} = 3$$

$$M(5,3) = \max \{M(2, 2)+1, M(5, 2)\} = 3$$

$$M(5,4) = \max \{M(1, 3)+3, M(5, 3)\} = 3$$

Esempio

<i>i</i>		0	1	2	3	4	<i>j</i>
-	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	1	1	1	1	1	
3	0	1	2	2	2	2	
4	0	1	2	2	3	3	
5	0	1	3	3	3	3	
6	0	1	3	3	4	4	

$$\begin{aligned} \max x_1 + 2x_2 + x_3 + 3x_4 \\ 2x_1 + 3x_2 + 3x_3 + 4x_4 \leq 6 \\ x \in \{0,1\}^4 \end{aligned}$$

$$M(i, j) = \max \{M(i - b_j, j-1) + c_j, M(i, j-1)\}$$

$$M(6,1) = \max \{M(4, 0)+1, M(6, 0)\} = 1$$

$$M(6,2) = \max \{M(3, 1)+2, M(6, 1)\} = 3$$

$$M(6,3) = \max \{M(3,2)+1, M(6, 2)\} = 3$$

$$M(6,4) = \max \{M(2, 3)+3, M(6, 3)\} = 4$$

→ Valore della soluzione ottima

Per trovare la soluzione: nell'ultimo max abbiamo scelto $M(2, 3)+3$, cioè preso oggetto $\{4\}$, nel max per ottenere $M(2, 3)$ avevamo scelto $M(2, 1)$ cioè non abbiamo preso oggetti, nel max per ottenere $M(2, 1)$ avevamo scelto $M(0, 0)+1$ cioè preso $\{1\}$: la soluzione trovata è $\{1,4\}$

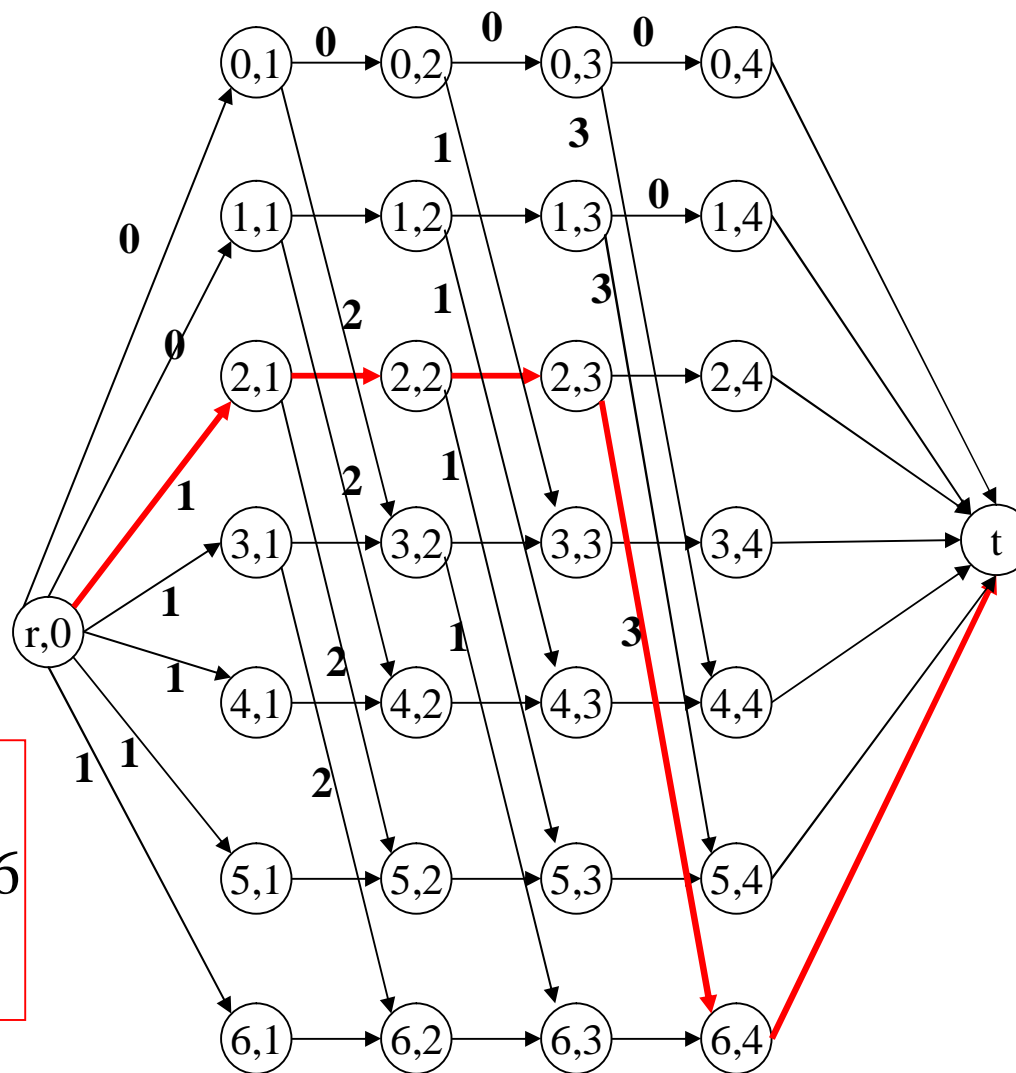
Il grafo dinamico associato

$$M(i, j) = \max \{M(i - b_j, j-1) + c_j, M(i, j-1)\}$$

Problema di cammino
massimo su grafo aciclico

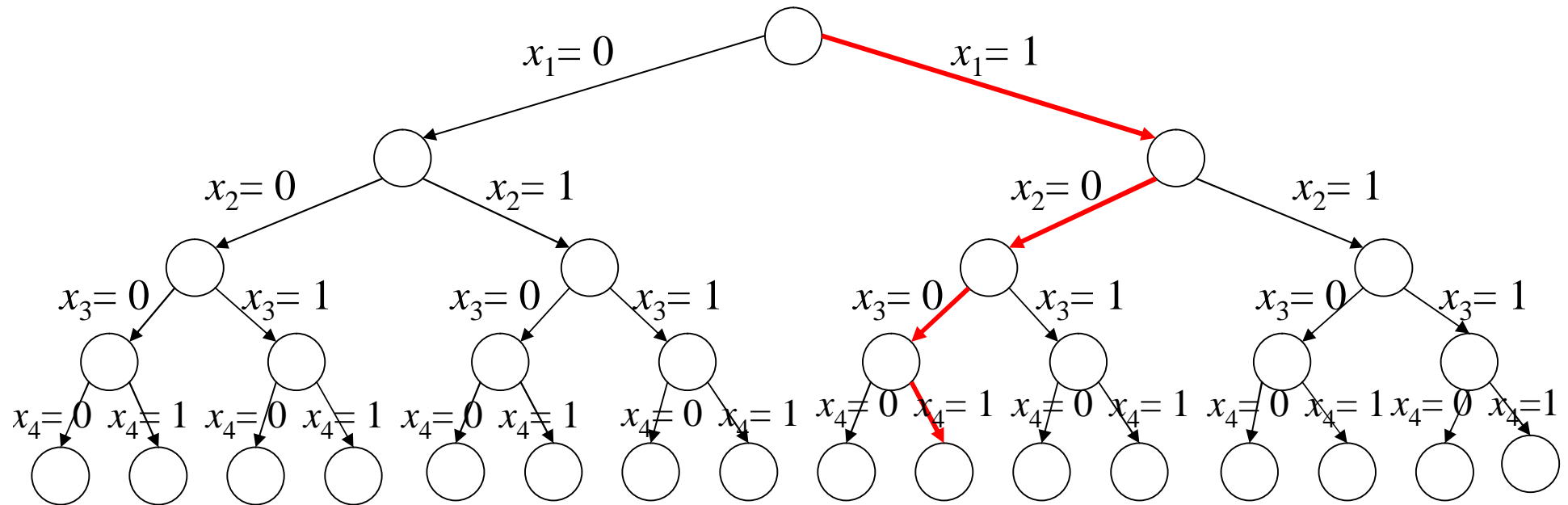
Gli archi senza
etichetta si intendono 0

Gli archi con un costo
 c_j rappresentano la
scelta dell'oggetto $\{j\}$



$$\begin{aligned} \max \quad & x_1 + 2x_2 + x_3 + 3x_4 \\ & 2x_1 + 3x_2 + 3x_3 + 4x_4 \leq 6 \\ & x \in \{0,1\}^4 \end{aligned}$$

Confronto con l'albero di branching



Numero di nodi 2^n : al crescere delle dimensioni del problema cresce più rapidamente!