Project Scheduling: CPM least cost scheduling

1. Introduzione

- *Tempo* e *costo* sono due degli aspetti più importanti in un progetto.

- Il *CPM least cost scheduling* è una tecnica che consente di considerare contemporaneamente questi due aspetti e risolvere il *problema del trade-off tempo-costo di un progetto*, assumendo che le durate delle attività non siano fissate a priori ma che possano assumere un valore compreso tra un minimo e un massimo.

In particolare:

- Il *CPM least cost scheduling* è una tecnica che che permette di *pianificare* la *riduzione* del *tempo di esecuzione* di un *progetto* con il *minimo aumento dei costi*, agendo sulle durate delle attività.

Project Scheduling: CPM least cost scheduling

2. Assunzioni di base

- Ciascuna attività può avere una durata compresa tra una durata minima e una massima.

- La *durata* in generale è *funzione* del *numero* e *tipo* di *risorse impiegate*.

- La *durata massima* viene definita dal project manager in base alle risorse a disposizione in *condizioni normali*: la durata massima rappresenta quindi la *durata normale*.

- La *durata minima* è la stima del tempo impiegato per eseguire l'attività nel modo più veloce possibile rispetto alle risorse in più che si possono utilizzare per l'attività (straordinari, secondo turno, macchine più efficienti, ecc.): la durata minima è detta *durata crash*.

- La modifica sulle durate delle attività influisce sulla *durata* e sui *costi* del progetto.

Project Scheduling: CPM least cost scheduling

2. Assunzioni di base (continua)

- Si considerano due tipi di costi:
  - *costi diretti*:
    direttamente attribuibili alle attività, come costi per l'uso di specifiche risorse, straordinari, secondo turno, macchine più efficienti, ecc.
  - *costi indiretti*:
    includono *costi non attribuibili* direttamente *alle* singole *attività*, come ad esempio costi di affitto attrezzature e locali, costo del personale.
    Di solito tali costi sono lineari rispetto al tempo.
- A questi possono aggiungersi eventuali:
  - *costi di penalità*:
    costi non attribuibili alle singole *attività*, e presenti ad esempio nel caso in cui non si rispettano le date di consegna stabilite contrattualmente per la conclusione del progetto o per l'esecuzione di parte del progetto stesso.
    Di solito tali costi sono lineari rispetto allo scarto (se positivo) tra il tempo di completamento e la data di consegna.
- Il *CPM least cost scheduling* focalizza l'attenzione sui *costi diretti*

Project Scheduling: CPM least cost scheduling

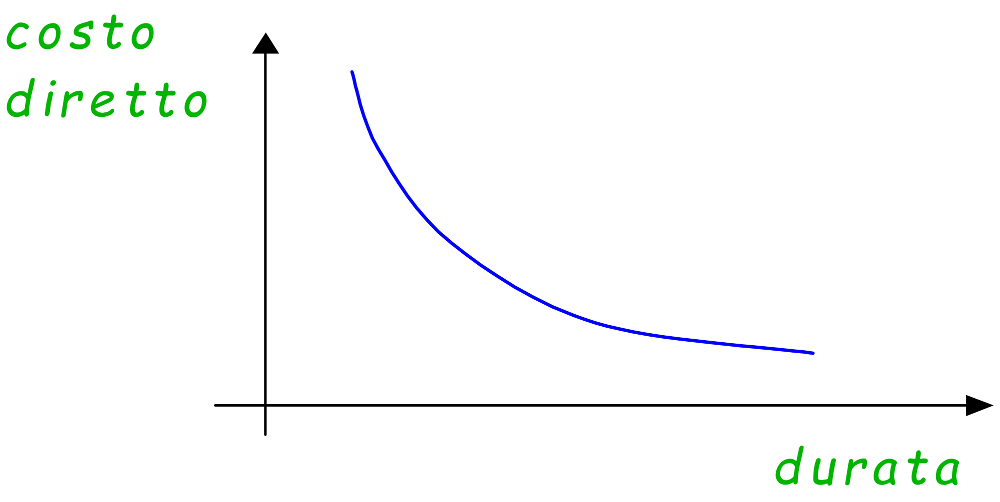
2. Assunzioni di base (continua)

- Il costo (diretto) della maggior parte delle attività dipende fortemente dal tempo impiegato per la loro esecuzione.

- Si ipotizza che i tempi si possano accorciare aumentando i costi (*straordinari, secondo turno, macchine più efficienti, ecc.*).

- I *costi diretti* chiaramente *aumentano* se viene *ridotta* la *durata* di *un'attività*, in quanto aumentano le risorse richieste o i requisiti sulla loro efficienza e in definitiva aumenta il relativo costo.

- Tipicamente tali costi hanno un *andamento convesso* rispetto al tempo.



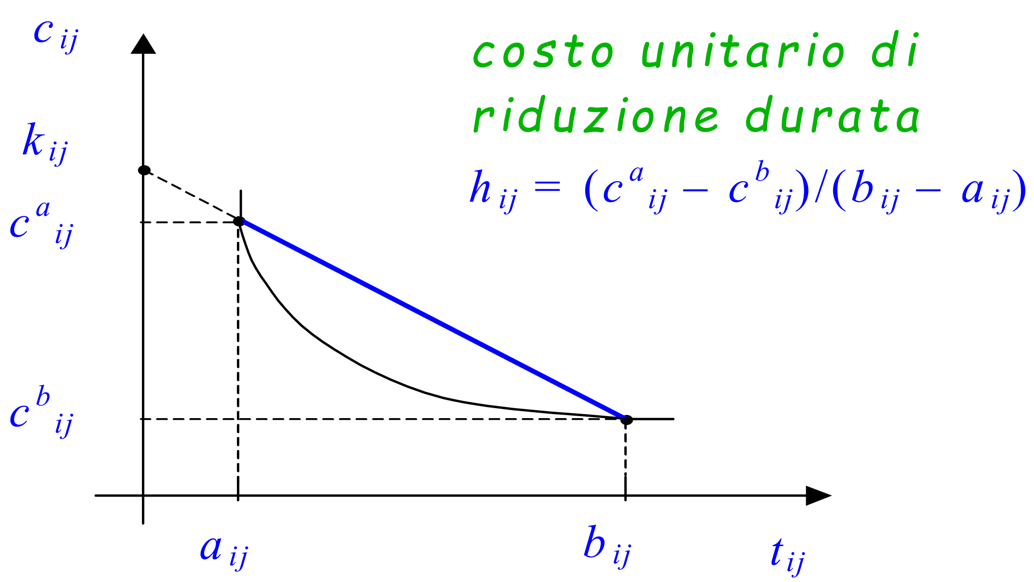
Project Scheduling: CPM least cost scheduling

2. Assunzioni di base (continua)

- Si considera un'*approssimazione lineare* della *curva tempo-costo* di un'attività.

- Si ipotizza cioè una relazione di tipo *lineare* tra i costi diretti c_{ij} e la durata t_{ij} di una generica attività (i, j) del tipo:

$$c_{ij} = k_{ij} - h_{ij}\, t_{ij}, \qquad \text{per} \quad a_{ij} \leq t_{ij} \leq b_{ij}$$



dove:

b_{ij} è la *durata normale* (massima),

a_{ij} è la *durata "crash"* (minima),

t_{ij} è la *durata effettiva*.

Project Scheduling: CPM least cost scheduling

2. Assunzioni di base (continua)

Osservazione 1

- Per ridurre il tempo di esecuzione del progetto occorre *agire* solo sulle *attività* dei *percorsi critici*.

Osservazione 2

- È inutile ridurre i tempi delle attività critiche al di sotto di valori per cui altri percorsi diventano critici e determinano pertanto la durata del progetto.

Osservazione 3

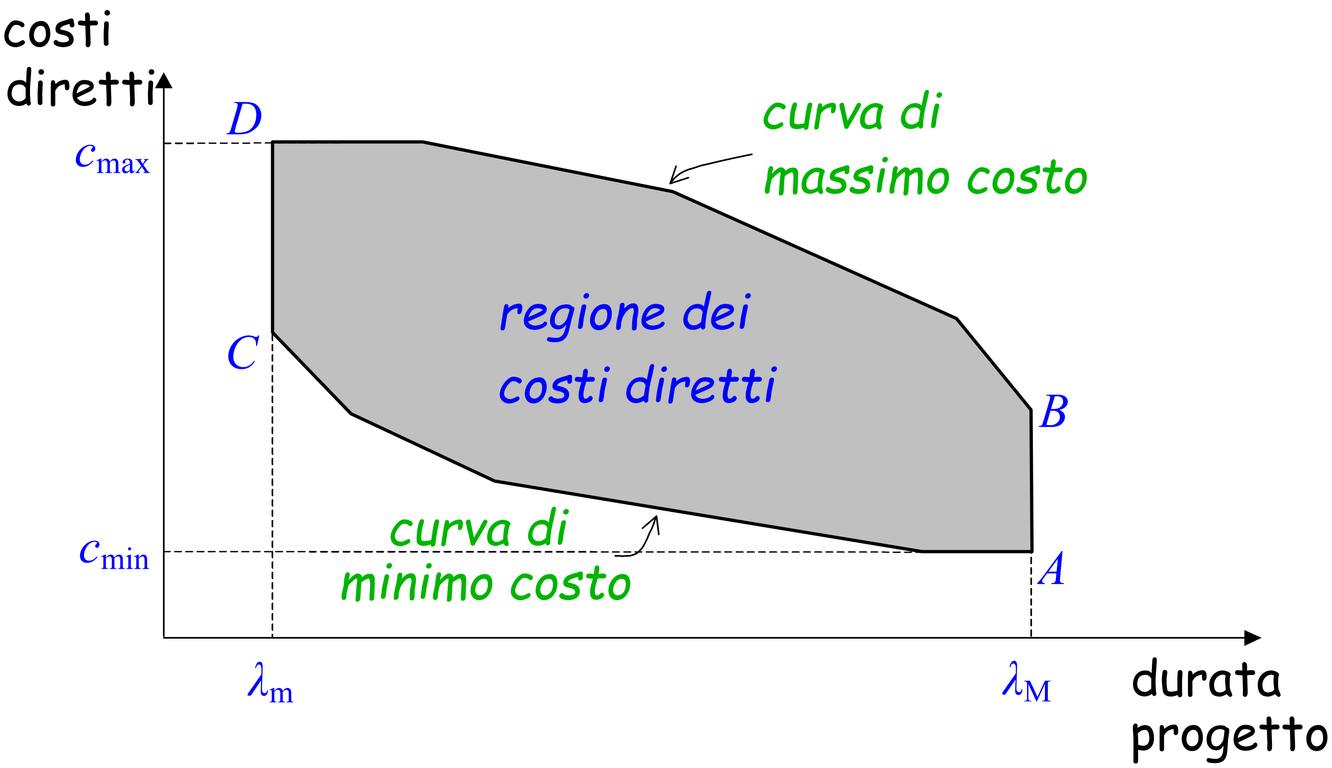
- Se esistono più percorsi critici, per ridurre la durata del progetto occorre *ridurre contemporaneamente* i tempi di *tutti i percorsi critici*.

- Il *CPM least cost scheduling* permette *ridurre* il *tempo* di *esecuzione* di un *progetto* con il *minimo aumento* dei *costi*, agendo solo sulle durate di quelle *attività* del *percorso critico* che presentano *costo unitario* totale *di riduzione* h *minimo*.

Project Scheduling: CPM least cost scheduling

3. Regione ammissibile

- È possibile far riferimento ad una *regione ammissibile* di punti nel piano *tempo-costi* che rappresenta tutte le possibili coppie (durata, costi diretti) per un progetto.



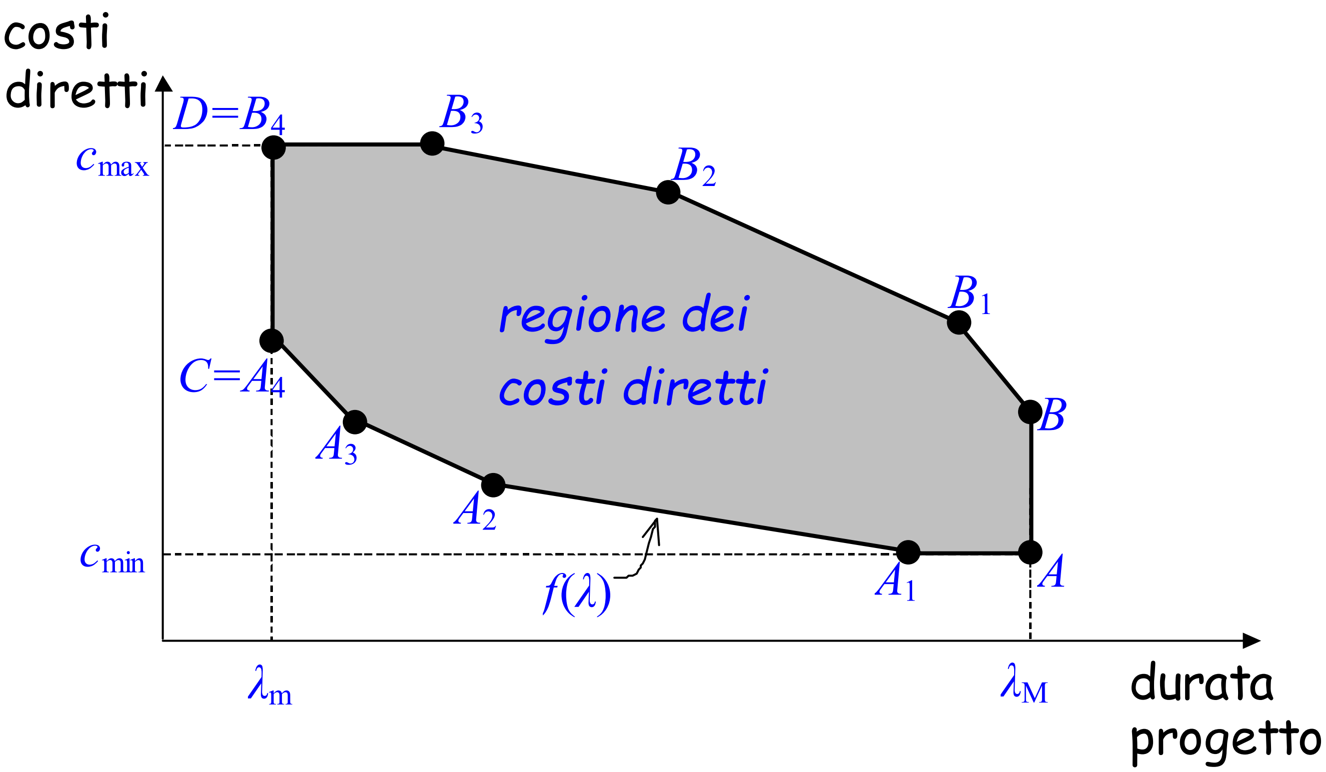
- Il punto A rappresenta la situazione in cui *ogni attività* è a *durata normale*.

- Il punto D rappresenta la situazione in cui *ogni attività* è a *durata crash*.

Project Scheduling: CPM least cost scheduling

3. Regione ammissibile (continua)

- Dato un progetto la progressiva riduzione a costo minimo della (minima) durata λ del progetto è rappresentata come in figura dalla curva A, A_1, A_2, A_3, A_4.



- Il punto A rappresenta il *massimo* valore del *tempo* (minimo) di *completamento* del progetto a *minimo costo* diretto.

- Il punto B rappresenta il *massimo* valore del *tempo* (minimo) di *completamento* del progetto a *massimo costo* diretto.

Project Scheduling: CPM least cost scheduling

3. Regione ammissibile (continua)

- Il punto $C=A_4$ rappresenta il *minimo* valore del *tempo* (minimo) di *completamento* del progetto a *minimo costo* diretto.

- Per arrivare al punto C occorre ridurre al minimo i tempi di tutte le attività del progetto che appartengono ai percorsi critici che vengono via via individuati.

- Se a partire da A_4 si riducessero al minimo i tempi di tutte le attività si avrebbe solo un aumento dei costi (punto $D = B_4$).

- Il punto $D=B_4$ rappresenta il *minimo* valore del *tempo* (minimo) di *completamento* del progetto a *massimo costo* diretto.

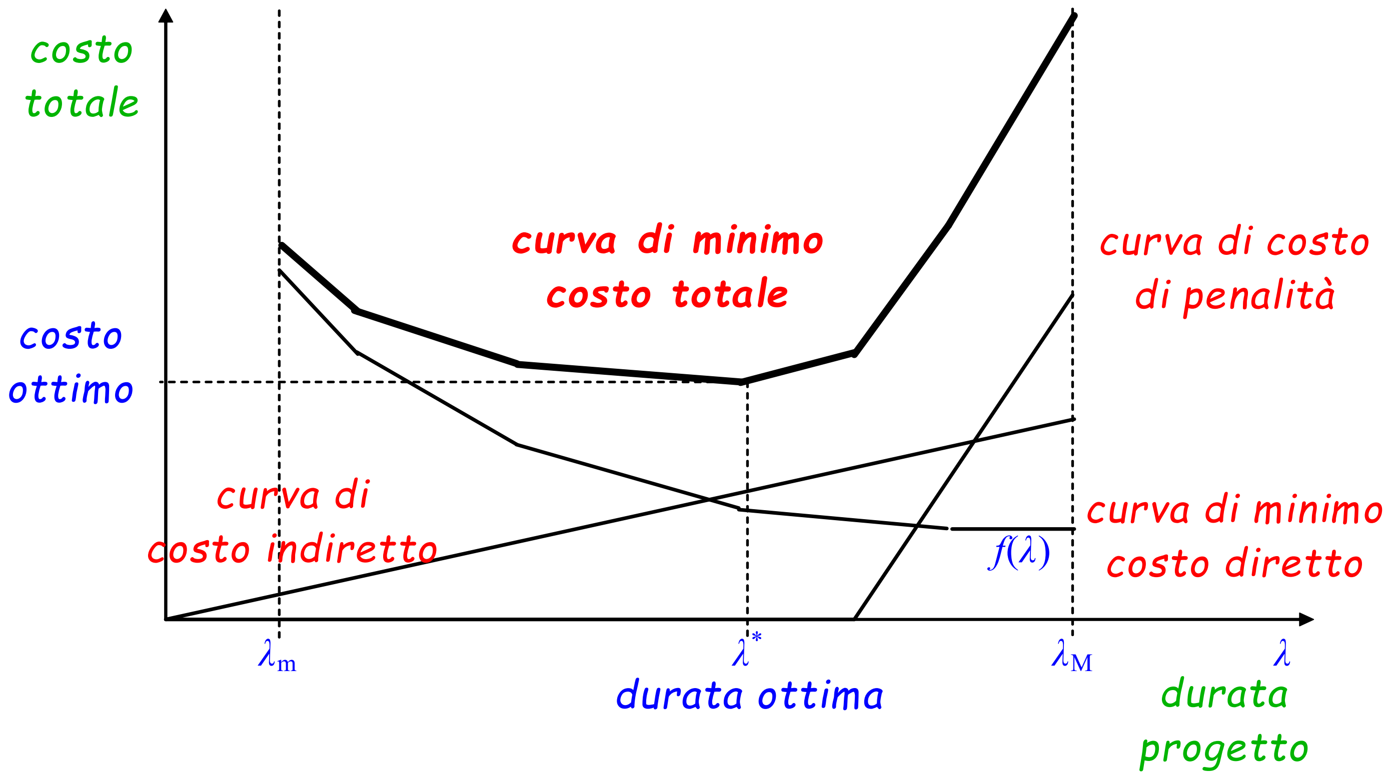
- Analogamente il punto B rappresenta il costo che il progetto avrebbe riducendo al minimo il tempo di tutte le attività tranne quelle critiche.

- Il problema che il *CPM least cost scheduling* risolve consiste nella *determinazione* della *curva tempo-costo minimo* $f(\lambda)$ che delimita inferiormente tale regione ammissibile.

Project Scheduling: CPM least cost scheduling

3. Regione ammissibile (continua)

- Una volta nota la *curva di minimo costo* è possibile definire la curva di costo totale minimo (diretto + indiretto + penalità) e quindi risalire alla *durata ottimale* del progetto per cui il *costo totale* del progetto è *minimo*



Project Scheduling: CPM least cost scheduling

4. Curva di minimo costo

- La *curva* $f(\lambda)$ ha un *andamento convesso* (nelle ipotesi fatte sui costi delle attività) ed in particolare *lineare a tratti*.

- Detto λ il tempo di completamento del progetto, con $\lambda_m \leq \lambda \leq \lambda_M$, il costo minimo complessivo corrispondente è:

$$f(\lambda) = \min_{a_{ij} \leq t_{ij} \leq b_{ij}} \{\Sigma_{i,j}\, k_{ij} - \Sigma_{i,j}\, h_{ij}\, t_{ij}\!: t_f - t_0 = \lambda\}$$
$$\text{tale che} \quad t_i + t_{ij} - t_j \leq 0 \quad \forall\ (i, j) \in \mathcal{A}$$

dove:

$$f(\lambda_m) \leq \Sigma_{i,j}\, k_{ij} - \Sigma_{i,j}\, h_{ij}\, a_{ij}$$

$$f(\lambda_M) = \Sigma_{i,j}\, k_{ij} - \Sigma_{i,j}\, h_{ij}\, b_{ij}$$

- Posto:

$$g(\lambda) = f(\lambda) - f(\lambda_M),$$

il *problema* si riduce a *trovare la funzione* $g(\lambda)$ che misura solo gli incrementi di costo rispetto al valore minimo $f(\lambda_M)$.

Si ha quindi $g(\lambda_M) = 0$.

Project Scheduling: CPM least cost scheduling

5. Procedura di base

- Sia data per la generica attività (i, j) una scelta del tempo t_{ij} di esecuzione, con:

$$a_{ij} \leq t_{ij} \leq b_{ij}$$

- La differenza $\varepsilon_{ij} = (t_{ij} - a_{ij})$ è il *margine disponibile* per la *riduzione* della durata dell'attività (i, j).

- Supponiamo che nel progetto ci sia un unico percorso (catena) critico $C_{\bar{\alpha}}$, cioè tale che $\ell_{\bar{\alpha}} > \ell_{\alpha}$, $a \in \{1, \ldots, p\}\backslash\{\bar{a}\}$

- l'*attività* (p, q) per cui è possibile una riduzione della durata a *minimo costo unitario* è quella per cui

$$h_{pq} = \min \{h_{ij} \,|(i, j) \in C_{\bar{\alpha}},\, \varepsilon_{ij} > 0\}$$

  dove h_{ij} è il *costo diretto unitario di riduzione* della durata dell'attività (i, j).

- Agendo su (p, q) è possibile ottenere una prima riduzione del tempo di completamento t_f del progetto a minimo costo.

Project Scheduling: CPM least cost scheduling

5. Procedura di base (continua)

- L'*entità* della *riduzione* della *durata* di un'attività critica (p, q) è *limitata* da *due fattori*:
  - (i) il *margine disponibile* ε_{pq};
  - (ii) la condizione che i *percorsi critici contenenti l'attività critica* in esame *rimangano critici*.

- Pertanto la *massima riduzione* possibile della durata t_{pq} dell'attività (p, q) è data da:

$$\Delta t^{*}=\max\{\Delta t |\ \Delta t \leq \varepsilon_{pq},\ (t_f - \Delta t) \geq l_\alpha,\ \forall\ \alpha\text{: } (p, q) \notin C_\alpha\}$$

- Il *costo* della *riduzione* della durata del progetto da t_f a $t_f - \Delta t^{*}$ è:

$$\Delta z = h_{pq}\ \Delta t^{*}$$

- Se i percorsi critici sono più di uno allora la riduzione di t_f a costo unitario minimo si ottiene *riducendo contemporaneamente* e nella stessa misura i tempi di *attività* situate su diversi *percorsi critici*.

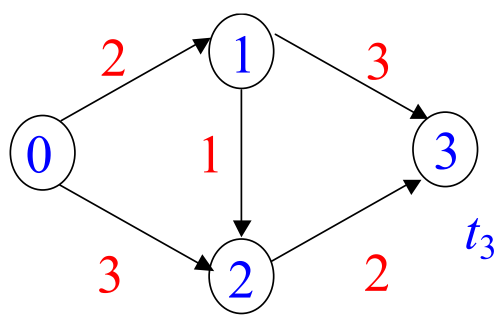
6. *Riduzione multipla*

- Quando si riducono i tempi di più attività contemporaneamente *può verificarsi* che alcuni percorsi subiscano una *riduzione multipla*.

- Il costo complessivo minimo si ottiene bilanciando possibilmente questa riduzione multipla con un opportuno aumento del tempo di esecuzione delle attività per le quali questo può avvenire senza incidere su t_f.

Esempio

- Consideriamo la rete seguente a cui si è arrivati dopo opportune riduzioni sulle durate delle attività.



attività	a_{ij}	t_{ij}	h_{ij}
(0, 1)	1	2	3
(0, 2)	3	3	1
(1, 2)	1	1	2
(1, 3)	3	3	1
(2, 3)	1	2	3

- N.B.: tutti i percorsi sono critici.

- Dalla tabella si nota che si può ottenere una ulteriore riduzione di t_3 riducendo di 1 i tempi sulle attività (0, 1) e (2, 3). In questa ipotesi $t_3 = 4$ ed il corrispondente incremento di costo è $\Delta c = 3+3= 6$.

Project Scheduling: CPM least cost scheduling

6. Riduzione multipla (continua)

- Con le riduzioni apportate il *percorso* (0, 1, 2, 3) *non* risulta più *critico* in quanto ad esso appartengono entrambe le attività (0, 1) e (2, 3).

- Su tale *percorso* si ha quindi una *riduzione doppia*.

- Ciò comporta un *non necessario incremento* di *costo*.

- Questo si può *compensare incrementando* di 1 la durata dell'attività (1, 2) ponendo cioè $t_{12} = 2$.

- In tal modo il *percorso* (0, 1, 2, 3) *torna* ad essere *critico* e l'incremento di costo diventa $\Delta c = 3 + 3 - 2 = 4$, riducendosi così del 33%.

Project Scheduling: CPM least cost scheduling

7. Formulazione come problema di PL

- Il problema della *determinazione* di $g(\lambda)$ può essere formulato come *problema di PL*.

- Infatti

$$g(\lambda) = f(\lambda) - f(\lambda_M) = \min_{a_{ij} \leq t_{ij} \leq b_{ij}} \{-\Sigma_{i,j}\, h_{ij}\, t_{ij};\ t_f - t_0 = \lambda\} + \Sigma_{i,j}\, h_{ij}\, b_{ij},$$

s.t.

$$t_i + t_{ij} - t_j \leq 0 \quad \forall\ (i, j) \in \mathcal{A}$$

- Occorre quindi risolvere il seguente PL:

$$\max\ z(\lambda) = \Sigma_{i,j}\, h_{ij}\, t_{ij}$$

s.t.

$$
\begin{array}{ll}
t_i + t_{ij} - t_j \leq 0 & \forall\ (i, j) \in \mathcal{A} \\
t_{ij} \leq b_{ij} & \forall\ (i, j) \in \mathcal{A} \\
-t_{ij} \leq -a_{ij} & \forall\ (i, j) \in \mathcal{A} \\
t_f - t_0 \leq \lambda & \\
t_i \gtrless 0 & \forall\ i \in \mathcal{N}
\end{array}
$$

dove $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ è la rete di progetto con attività sugli archi.

Project Scheduling: CPM least cost scheduling

7. Formulazione come problema di PL (continua)

Osservazione 1

- Detti λ_m e λ_M i valori dei tempi minimi di completamento del progetto che si ottengono ponendo rispettivamente

$$t_{ij} = a_{ij} \quad \text{e} \quad t_{ij} = b_{ij} \quad \text{per ogni attività } (i, j),$$

allora il *problema* di PL *ha senso per*

$$\lambda_\mathrm{m} \leq \lambda \leq \lambda_\mathrm{M}.$$

- Infatti per $\lambda < \lambda_\mathrm{m}$ il problema non ha soluzioni ammissibili, mentre per $\lambda > \lambda_\mathrm{M}$ la soluzione ottima è data da $t_{ij} = b_{ij}$.

Osservazione 2

- È inoltre chiaro che il valore ottimo $z^*(\lambda)$ cresce al decrescere di λ, e per ogni assegnato valore di λ, con $\lambda_\mathrm{m} \leq \lambda \leq \lambda_\mathrm{M}$, all'ottimo risulta sempre

$$t_f - t_0 = \lambda.$$

Project Scheduling: CPM least cost scheduling

7. Formulazione come problema di PL (continua)

Osservazione 3

- Per quanto riguarda il primo vincolo, questo è una diretta conseguenza della definizione del tempo minimo di raggiungimento di un nodo:

$$t_i = \max_{\{h\,:\,(h,\,i)\,\in\,\mathcal{A}\}} \{t_h + t_{hi}\}$$

  ma non è equivalente ad essa.

- Infatti la soluzione del problema non garantisce che per ogni nodo i il vincolo stesso risulti soddisfatto all'uguaglianza per almeno un nodo i.

- Ciò non di meno, ai fini della determinazione dei valori ottimi t^*_{ij} che minimizzano il costo di un progetto con durata λ prefissata, il primo vincolo e la definizione di t_i sono equivalenti, nel senso che esiste almeno una soluzione ottima del problema in cui i valori dalle variabili t_i sono equivalenti a quelli dei tempi minimi di raggiungimento dei nodi con attività di durate t^*_{ij}.

Project Scheduling: CPM least cost scheduling

7. Formulazione come problema di PL (continua)

Osservazione 4

- Quindi, risolvendo il problema di PL formulato si può verificare che il valore di t_i risultante può non corrispondere al tempo minimo di raggiungimento del nodo i.

- Questo può essere calcolato successivamente una volta trovati i valori ottimi dei t_{ij}.

• In linea di principio, $g(\lambda)$ può essere ricavata per punti risolvendo il problema di PL per ogni $\lambda \in [\lambda_m, \lambda_M]$.

• Ipotizzando senza perdita di generalità valori interi per le durate crash e normali delle attività si può dimostrare che i valori ottimi del problema di PL sono interi (matrice dei vincoli totalmente uni modulare) e quindi è sufficiente ricavare $g(\lambda)$ per valori interi di λ.

• Inoltre, è noto che $g(\lambda)$ è lineare a tratti.

• E' quindi sufficiente risolvere il problema di PL per ogni $\lambda = \lambda_\ell$, dove λ_ℓ è l'estremo destro dell'ℓ-esimo sotto-intervallo di $[\lambda_m, \lambda_M]$ nel quale $g(\lambda)$ è lineare.

7. Formulazione come problema di PL (continua)

- In particolare, detti q i sottointervalli $[\lambda_{\ell\text{-}1}, \lambda_\ell]$ di $[\lambda_m, \lambda_M]$ con $\lambda_1 < \lambda_2 < \ldots < \lambda_\ell < \ldots < \lambda_q = \lambda_M$ (e $\lambda_0 = \lambda_m$) si risolvono i problemi di PL per valori decrescenti di λ_ℓ.
- Si noti che l'analisi di sensibilità della soluzione ottima del problema di PL per il generico $\lambda = \lambda_\ell$ consente di determinare:
  1. la pendenza del tratto lineare di $g(\lambda)$ nel sottointervallo $[\lambda_{\ell\text{-}1}, \lambda_\ell]$, ricavabile dal prezzo-ombra (valore variabile duale) associato al vincolo di durata del progetto;
  2. la larghezza del sottointervallo $[\lambda_{\ell\text{-}1}, \lambda_\ell]$, pari alla larghezza dell'intervallo di valori del parametro λ al cui interno il prezzo-ombra associato al vincolo di durata del progetto resta invariato.
- Il problema di PL prima formulato si può risolvere applicando gli algoritmi di programmazione lineare (*metodo del simplesso*).
- Tuttavia, a causa delle notevoli dimensioni che spesso assume il corrispondente programma si può ricorrere ad *algoritmi* specifici che sfruttano i concetti della teoria dei *flussi su rete*.
- Nel seguito descriveremo *un approccio risolutivo* per determinare $g(\lambda)$ basato sulla teoria delle reti di flusso.

Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete

- $g(\lambda)$ viene descritta attraverso un *metodo iterativo* in cui *ad ogni iterazione* si *individua* l'insieme Θ di *attività che consentono*, variandone la durata, una *riduzione unitaria della durata del progetto a minimo costo*.

- Chiaramente l'insieme Θ è *contenuto* nell'insieme delle *attività critiche*.

- Si parte dalla situazione in cui tutte le *attività* sono alla loro *durata normale*, cioè $t_{ij} = b_{ij}$ per ogni $(i, j) \in \mathcal{A}$

$$f(\lambda_M) = \Sigma_{i,j}\, k_{ij} - \Sigma_{i,j}\, h_{ij}\, b_{ij}, \qquad\qquad g(\lambda_M) = 0,$$

- La durata minima del progetto è λ_M e $g(\lambda_M) = 0$.

- Si modificano le durate delle attività in Θ della *massima variazione possibile* che consente di *ridurre* della stessa quantità tutti i *percorsi critici*, garantendo di preservarne la criticità.

- Il procedimento *termina* quando su almeno uno dei percorsi critici non è possibile effettuare alcuna riduzione.

Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

- Il cuore della procedura è chiaramente *individuare l'insieme Θ di attività*.
- Questo problema corrisponde a risolvere un problema di *massimo flusso sulla sottorete indotta dai soli cammini critici*.
- Ad ogni iterazione si ipotizza di partire da una *schedula a costo minimo (MCS)* per una data durata λ del progetto.
- La *MCS* è associata ad un insieme di durate individuali per ogni attività in modo tale che la durata corrispondente λ del progetto non possa essere ottenuta a costo minore attraverso un altro insieme di durate ammissibili per le attività.

Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

- All'inizio supporremo che tutte le attività siano alla loro durata normale a cui corrisponde durata λ_M per la *MCS* del progetto e costo diretto minimo $f(\lambda_M)$.

- Siamo interessati ad ottenere una nuova *MCS* di durata ridotta di una unità di tempo.

- È evidente che non andremo a modificare attività non critiche.

- Pertanto considereremo solo la *sottorete* consistente delle sole *attività critiche*.

- Siano $A_1, \ldots, A_\mu$, con $\mu \leq m$, le *attività critiche* (m: numero di attività del progetto (archi della rete di progetto)).

Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

- Siano P_i, per $i = 1, ..., k$, tutti i *cammini* (*critici*) dal nodo iniziale (sorgente) 0 a quello finale (pozzo) f della *sottorete critica*,

- Sia

$$a_{ij} = \begin{cases} 1 & \text{se } A_j \in P_i \\ 0 & \text{altrimenti} \end{cases}$$

  La matrice $\{a_{ij}\}$ è la *matrice* di *incidenza cammini-archi*.

- Definiamo con x_j la variabile reale corrispondente alla *riduzione della durata dell'attività* A_j.

- Pertanto:
  - per un'attività A_j a durata *normale*, $x_j \geq 0$;
  - per un'attività A_j a durata *intermedia*, $x_j \gtrless 0$;
  - per un'attività A_j a durata *crash*, $x_j \leq 0$.

Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

- Il problema di determinare una nuova *MCS* di durata ridotta di una unità può formularsi tramite il seguente *problema* (P) (*primale*) di *PL*:

(P)
$$\min\ z = \sum_{j=1}^{n} h_j x_j$$
s.t.
$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, \quad \text{per tutti i cammini } P_i,\ i = 1, \ldots, k$$
$$x_j \begin{cases} \geq 0 & \text{se } A_j \text{ è a durata } \textit{normale} \\ \gtrless 0 & \text{se } A_j \text{ è a durata } \textit{intermedia} \\ \leq 0 & \text{se } A_j \text{ è a durata } \textit{crash} \end{cases} \quad j = 1, \ldots, n$$

- Consideriamo il *problema duale* (D):

(D)
$$\max\ w = \sum_{i=1}^{k} y_i$$
s.t.
$$\sum_{i=1}^{k} a_{ij} y_i \begin{cases} \leq h_j & \text{se } A_j \text{ è a durata } \textit{normale} \\ = h_j & \text{se } A_j \text{ è a durata } \textit{intermedia} \\ \geq h_j & \text{se } A_j \text{ è a durata } \textit{crash} \end{cases} \quad j = 1, \ldots, n$$
$$y_i \geq 0, \qquad i = 1, \ldots, k$$

- Si noti che y_i può interpretarsi come il *flusso lungo il cammino* P_i, mentre i vincoli definiscono i *vincoli di capacità* (minima e massima) su ciascun arco A_j della sottorete critica.

Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

- Pertanto il *problema duale* (D) corrisponde al *problema del flusso massimo sulla sottorete* $\mathcal{R}_c = (\mathcal{N}_c, \mathcal{A}_c)$ *delle attività critiche* ($\mathcal{A}_c \subseteq \mathcal{A}$ e $\mathcal{N}_c = \{u \in \mathcal{N}: \exists\ (u, v) \text{ o } (v, u) \in \mathcal{A}_c\}$) in cui le capacità minime e massime sui flussi sugli archi $A_j = (u, v) \in \mathcal{A}_c$ sono definiti dai costi h_j.

- In particolare, detti u_{uv} e ℓ_{uv} la massima e minima capacità dell'arco (u, v) relativo all'attività $A_j = (u, v)$, si ha che:
  - $\ell_{uv} = 0$, $u_{uv} = h_j$ se A_j è a durata *normale*.
  - $\ell_{uv} = h_j$, $u_{uv} = h_j$ se A_j è a durata *intermedia*.
  - $\ell_{uv} = h_j$, $u_{uv} = +\infty$ se A_j è a durata *crash*.

  N.B.: le attività fittizie si assumono a durata normale e con costo di riduzione illimitato superiormente.

- È noto che il flusso massimo su una rete è pari alla minima capacità dei *tagli* s–t $[S, \bar{S}]$, con $s \in S \subset \mathcal{N}_c$ e $t \in \bar{S} = \mathcal{N}_c \backslash S$, dove la capacità $u[S, \bar{S}]$ del taglio s–t $[S, \bar{S}]$ è

$$u[S, \bar{S}] = \Sigma_{(u,\, v) \in (S,\, \bar{S})}\, u_{uv} - \Sigma_{(u,\, v) \in (\bar{S},\, S)}\, \ell_{uv}$$

Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

- Il *taglio di capacità minima* (finita), corrispondente al *flusso massimo*, permette di individuare l'insieme di archi, cioè l'insieme $\Theta = [S, \bar{S}]$ di *attività critiche per le quali occorre modificarne la durata*.

- Modificando le durate di tali attività si può *diminuire* la *durata* del *progetto* di una unità di tempo con il *minimo incremento di costo diretto*.

- L'*incremento di costo* (unitario) è pari alla *capacità del taglio minimo* individuato.

- Le durate delle attività corrispondenti al taglio vengono così modificate:

  (a) *per ogni attività corrispondente ad un arco in* $(S, \bar{S})$ *si riduce la durata*.
  (N.B.: tali attività non possono essere a durata crash altrimenti il taglio avrebbe capacità infinita);

  (b) *per ogni attività, non a durata normale, corrispondente ad un arco in* $(\bar{S}, S)$ *si aumenta la durata*.
  (N.B.: se l'attività è a durata normale il vincolo di capacità minima sul flusso dell'arco relativo, essendo pari a zero, non influisce sulla capacità del taglio).

8. Approccio basato su flusso su rete (continua)

- Queste operazioni complessivamente comportano un aumento del *costo diretto*, *per unità di tempo*, *pari* alla *capacità* $u[S, \bar{S}]$ del *taglio* s–t $[S, \bar{S}]$.

  Teorema

  - Siccome le operazioni (a) e (b) corrispondono ad una soluzione ammissibile per il problema (P) di valore pari a quello di una soluzione ammissibile del suo duale (D), ne segue che la soluzione è anche ottima.

- Se il problema di *massimo flusso* ha soluzione di valore *illimitato* $w^* = \infty$ (cioè non c'è alcun taglio $[S, \bar{S}]$ di capacità finita) allora la *durata del progetto non può essere ridotta* ulteriormente in quanto il problema (P) *non ammette soluzione*.

Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

Algoritmo CPM least cost scheduling

Passo 0. Si fissi il tempo effettivo di ogni attività alla sua durata normale.

Passo 1. Si effettui l'analisi dei tempi del progetto con le attività alla loro durata effettiva in modo da individuare i percorsi critici.

Passo 2a. Si faccia riferimento alla sottorete costituita dalle sole attività critiche $A_j = (u, v)$ e si associno agli archi (u, v) opportune capacità minime e massime secondo lo schema:

- $\ell_{uv} = 0$, $u_{uv} = h_j$ se A_j è a durata *normale*.
- $\ell_{uv} = h_j$, $u_{uv} = h_j$ se A_j è a durata *intermedia*.
- $\ell_{uv} = h_j$, $u_{uv} = \infty$ se A_j è a durata *crash*.

Si risolva il problema di massimo flusso sulla sottorete. In particolare, si individui il taglio $[S, \bar{S}]$ di capacità minima.

Sia $\bar{h}$ la capacità del taglio di capacità minima $[S, \bar{S}]$ e $\Theta = [S, \bar{S}]$ l'insieme delle attività relative a tale taglio.

Passo 2b. Se $\bar{h} = \infty$, stop.

Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

Passo 3. Si determini l'entità Δt della riduzione (aumento) delle attività relative agli archi in $(S, \bar{S})$ (in $(\bar{S}, S)$):

$$\Delta t = \min \{\Delta 1, \Delta 2\}$$

- $\Delta 1$ è la massima riduzione (aumento) possibile sulla base dei margini di riduzione (aumento) delle attività.
- $\Delta 2$ è la massima riduzione possibile per la quale i percorsi critici rimangano tali (cioè, la differenza tra la lunghezza del percorso critico e quella del percorso (subcritico) più lungo che non contiene le attività in Θ individuate al passo 2).

Passo 4. Si decrementi (aumenti) di Δt le durate delle attività relative agli archi in $(\bar{S}, S)$ (in (S, S)) e quindi di Δt la durata del progetto.

I costi diretti del progetto aumentano di

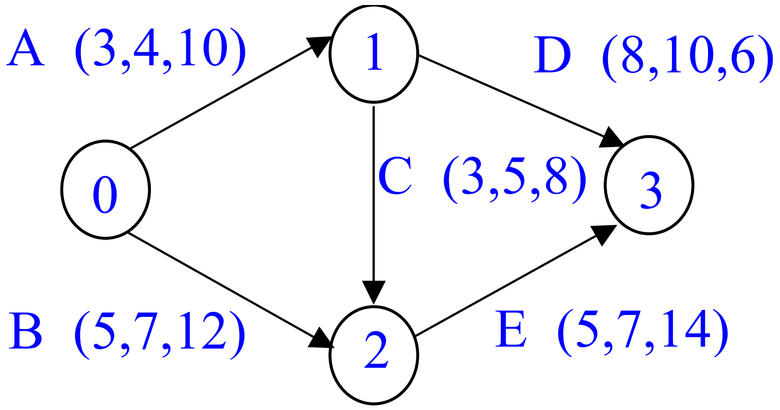
$$\Delta c = \Delta t\, \bar{h}$$

Si ritorni al Passo 1.

8. Approccio basato su flusso su rete (continua)

Esempio

- Si consideri il seguente progetto in cui si richiede l'esecuzione di cinque attività A, B, C, D, E, con le relazioni di precedenza: A < C, A < D, B, C < E.



(a_{ij},b_{ij},h_{ij})

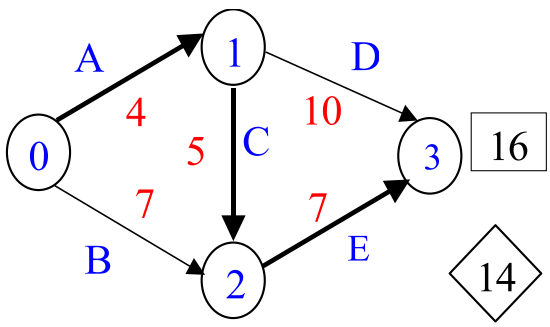
Analisi tempi (Critical path problem)

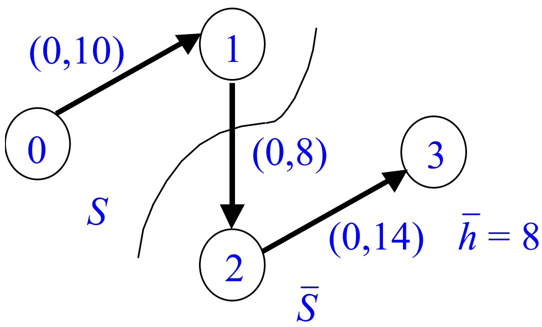
$\Delta t = \min\{\Delta 1, \Delta 2\}$
$\Delta c = \Delta t\, \bar{h}$

Massimo flusso sulla sottorete critica

(l_{ij}, u_{ij})

I)





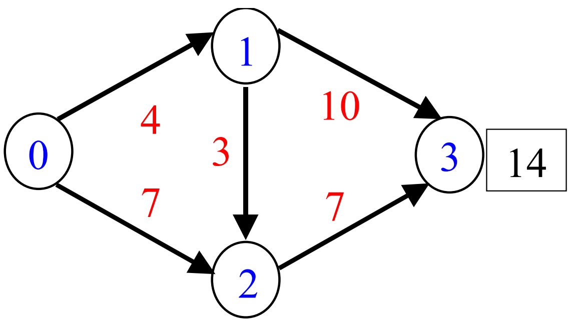
$$\Delta t = \min\{2, 2\} = 2$$
$$\Delta c = 2 \cdot 8 = 16$$

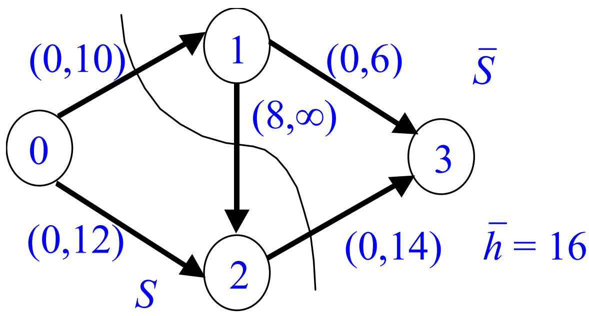
Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

Analisi tempi (Critical path problem)	$\Delta t = \min\{\Delta 1, \Delta 2\}$ $\Delta c = \Delta t \, \bar{h}$	Massimo flusso sulla sottorete critica

II)

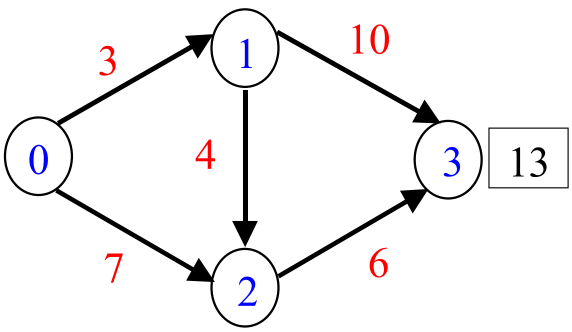


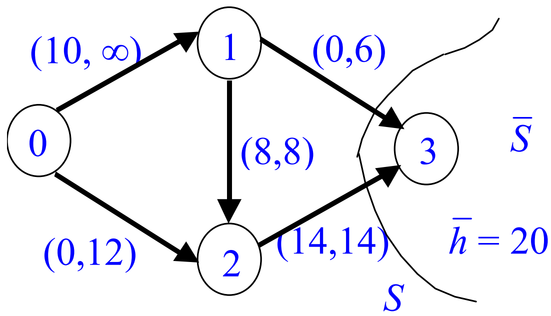


$\Delta t = \min\{1, -\} = 1$

$\Delta c = 1 \cdot 16 = 16$

III)

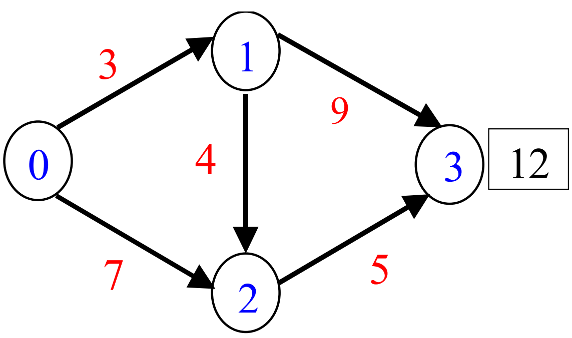


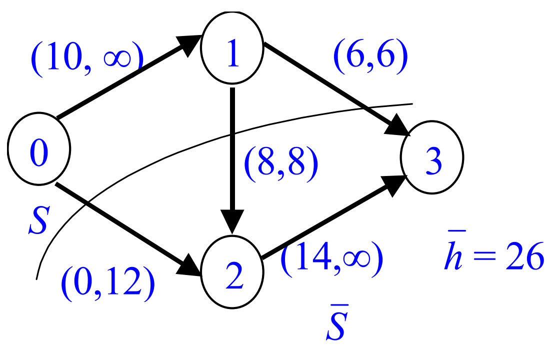


$\Delta t = \min\{1, -\} = 1$

$\Delta c = 1 \cdot 20 = 20$

IV)





$\Delta t = \min\{1, -\} = 1$

$\Delta c = 1 \cdot 26 = 26$

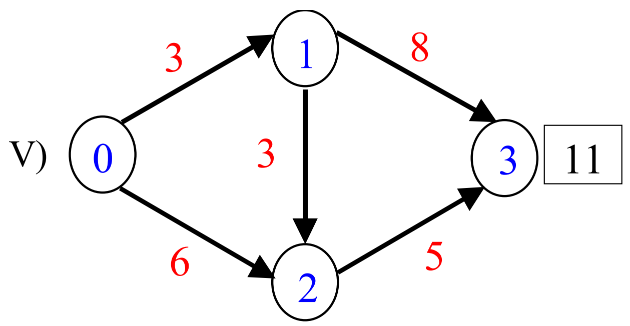
Project Scheduling: CPM least cost scheduling

8. Approccio basato su flusso su rete (continua)

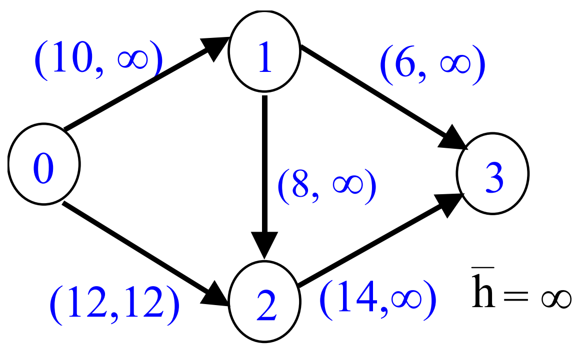
Analisi tempi
(Critical path problem)

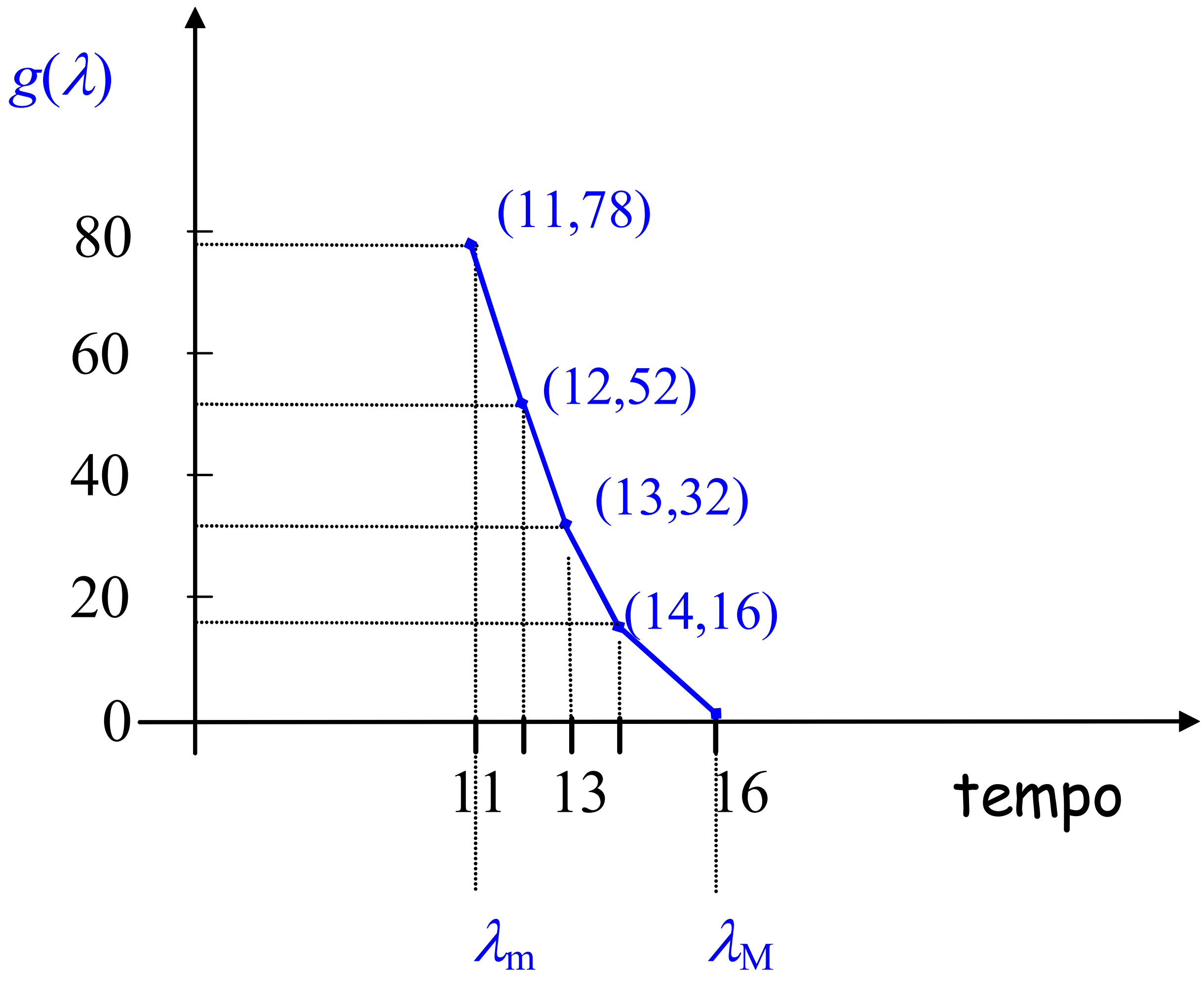
$\Delta t = \min\{\Delta 1, \Delta 2\}$
$\Delta c = \Delta t\, \bar{h}$

Massimo flusso
sulla sottorete critica



$\bar{h} = \infty$
STOP

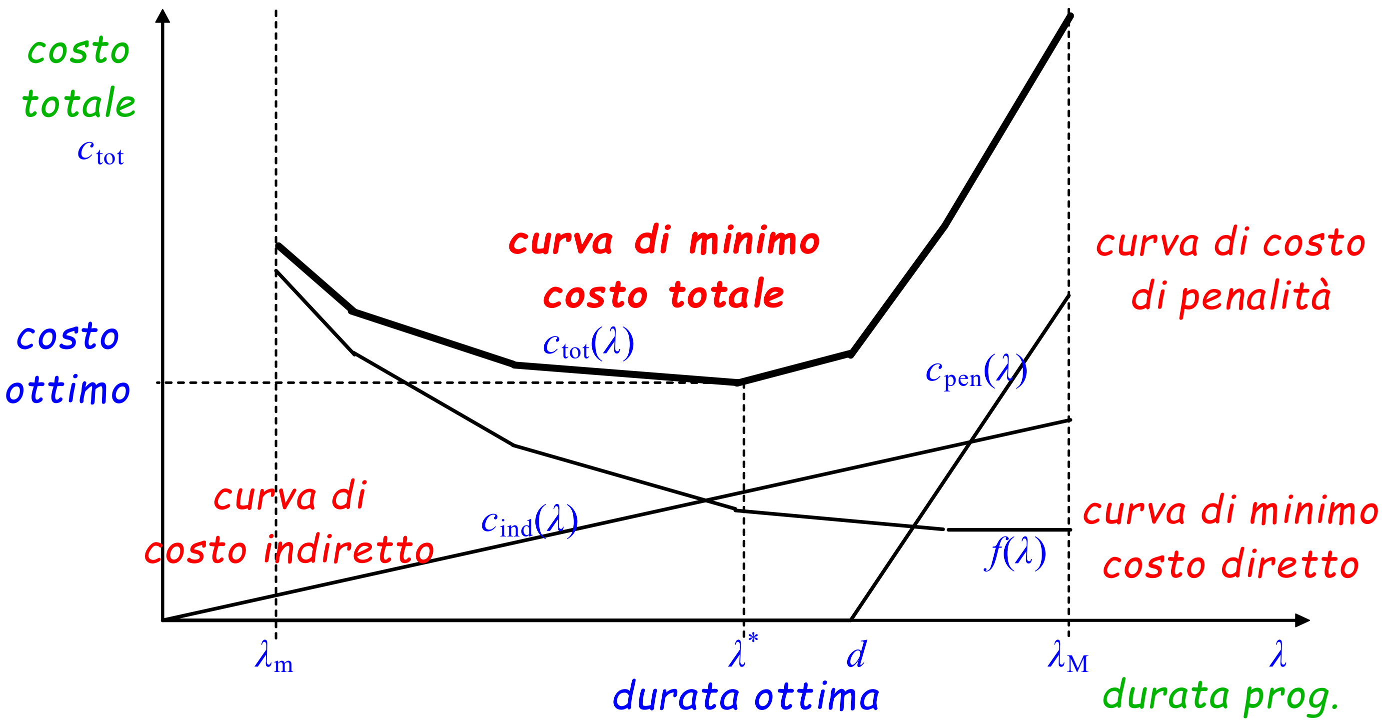




Project Scheduling: CPM least cost scheduling

9. Il caso con costi indiretti e di penalità

- Supponendo i costi indiretti e di penalità lineari la *curva di (minimo) costo totale* $c_{tot}(\lambda)$ (diretto + indiretto + penalità) al variare di λ è ancora lineari a tratti e convessa.



- In particolare siano $c_{ind}(\lambda) = c_0 \cdot \lambda$ il costo indiretto, e $c_{pen}(\lambda) = p_0 \cdot \max(0, \lambda - d)$ il costo di penalità rispetto alla (massima) data di fine presunta d del progetto.

- Quindi $c_{tot}(\lambda) = f(\lambda) + c_{ind}(\lambda) + c_{pen}(\lambda)$
$$= f(\lambda) + c_0 \cdot \lambda + p_0 \cdot \max(0, \lambda - d).$$

Project Scheduling: CPM least cost scheduling

9. Il caso con costi indiretti e di penalità (continua)

- Per determinare la *durata ottimale* λ^* del progetto per cui il *costo totale* dello stesso è *minimo* possiamo utilizzare l'algoritmo per determinare la funzione $f(\lambda)$, adattando il Passo 2b e il Passo 3 come segue, dove Δt è l'entità di riduzione della durata del progetto.

Passo 2b'. Se $\Delta c_{tot}/\Delta t \geq 0$, stop.

dove:

caso a) $\Delta c_{tot}/\Delta t = \bar{h} - c_0 - p_0$, se $\lambda > d$;

caso b) $\Delta c_{tot}/\Delta t = \bar{h} - c_0$, altrimenti.

Passo 3'. $\Delta t = \min \{\Delta 1, \Delta 2, \Delta 3\}$, con $\Delta 3 = \lambda - d$, se (*caso a*) $\lambda > d$; altrimenti (*caso b*) $\Delta t = \min \{\Delta 1, \Delta 2\}$.

- Nel caso in cui la massima data di fine d sia tassativa (*deadline*) le soluzioni ammissibili sono quelle con $\lambda \leq d$.

- Assumendo $p_0 = +\infty$, è possibile impiegare ancora lo stesso procedimento valido per il caso con costi di penalità (si noti che per $\lambda > d$, $\Delta c_{tot}/\Delta t = -\infty$).

- Alternativamente, considerando $c_{tot}(\lambda) = f(\lambda) + c_0 \cdot \lambda$, il Passo 2b' viene eseguito solo se $\lambda \leq d$ mentre al Passo 3' $\Delta t = \min \{\Delta 1, \Delta 2, \Delta 3\}$, se $\lambda > d$ e $\Delta c_{tot}/\Delta t = \bar{h} - c_0 \geq 0$; altrimenti $\Delta t = \min \{\Delta 1, \Delta 2\}$.

Project Scheduling: CPM least cost scheduling

9. Il caso con costi indiretti e di penalità (continua)

- Concludiamo formulando il problema della minimizzazione del costo totale in termini di PL:

$$
\begin{aligned}
\min\ & c_{\text{tot}} = \Sigma_{(i,j)\in\mathcal{A}}\,(c^b_{ij} + h_{ij}\,(b_{ij} - t_{ij})) + c_0\,\lambda + p_0\,T \\
\text{s.t.}\ & \\
& t_i + t_{ij} - t_j \le 0 && \forall\,(i,j) \in \mathcal{A} \\
& t_{ij} \le b_{ij} && \forall\,(i,j) \in \mathcal{A} \\
& t_{ij} \ge a_{ij} && \forall\,(i,j) \in \mathcal{A} \\
& t_f - t_0 \le \lambda && \\
& \lambda - d \le T && \\
& T \ge 0 && \\
& t_i \gtrless 0 && \forall\, i \in \mathcal{N}
\end{aligned}
$$

dove la f.o. può essere sostituita da quella equivalente

$$z = c_0\,\lambda + p_0\,T - \Sigma_{(i,j)\in\mathcal{A}}\,h_{ij}\,t_{ij},$$

avendo rimosso i termini costanti.

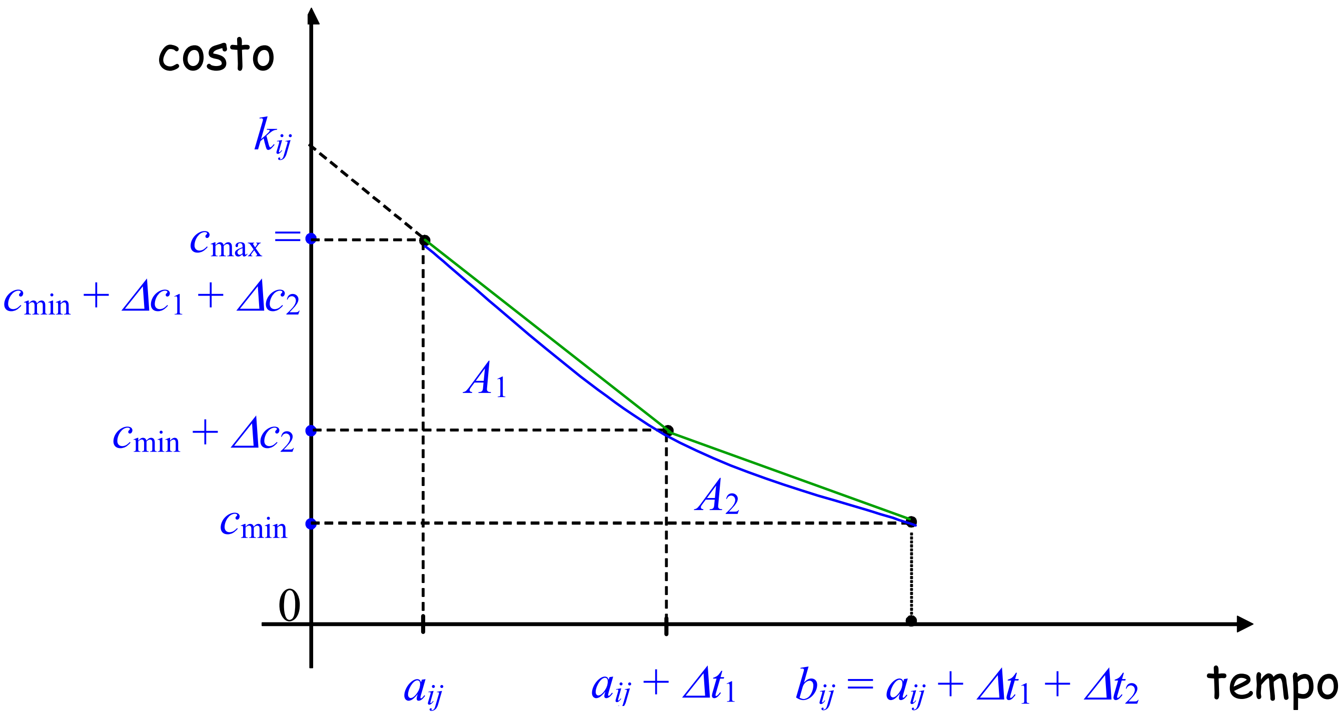
- È possibile ricondurre il problema (anche per il caso con deadline, cioè con $p_0 = +\infty$) ad un particolare problema di flusso a costo minimo su una rete non capacitata.

Project Scheduling: CPM least cost scheduling

10. Costi non lineari

- Consideriamo il caso di relazioni *tempo-costo non lineari*.

- In particolare, consideriamo un'attività A la cui *relazione tempo-costo* sia una funzione *continua*, *non crescente* e *convessa*, come in figura.

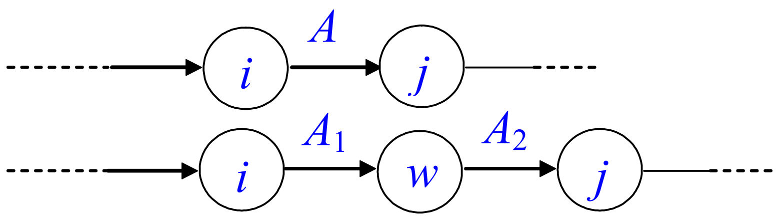


- Questa funzione di costo può essere *approssimata* con una *curva lineare a tratti*.

Project Scheduling: CPM least cost scheduling

10. Costi non lineari *(continua)*

- Si introduce un nodo fittizio w e quindi si *divide* l'attività A in due *pseudo-attività* A_1, A_2



$$i < w < j$$

- Ad A_1 e A_2 si fa corrispondere una *relazione tempo-costo lineare* del tipo:

(i) Attività $A_1 = (i, w)$

$$c_{iw} = k_{iw} - h_{iw}\, t_{iw}, \qquad a_{iw} \le t_{iw} \le b_{iw}$$

con $k_{iw} = k_{ij}$; $h_{iw} = \Delta c_1/\Delta t_1$; $a_{iw} = a_{ij}$, $b_{iw} = a_{ij} + \Delta t_1$

(ii) Attività $A_2 = (w, j)$

$$c_{wj} = k_{wj} - h_{wj}\, t_{wj}, \qquad a_{wj} \le t_{wj} \le b_{wj}$$

con $k_{wj} = 0$, $h_{wj} = \Delta c_2/\Delta t_2$; $a_{wj} = 0$, $b_{wj} = \Delta t_2$

Project Scheduling: CPM least cost scheduling

10. Costi non lineari (continua)

- Nella formulazione di pag. 16 (e in quella di pag. 36) le grandezze caratteristiche delle attività A_1 e A_2 compaiono nella f.o. e nei vincoli nel seguente modo:

$$\max \quad z(\lambda) = \cdots + h_{iw}\, t_{iw} + h_{wj}\, t_{wj} + \cdots$$

s.t.

$$\begin{aligned}
&\bullet\\
&\bullet\\
&t_i + t_{iw} - t_w \leq 0\\
&t_w + t_{wj} - t_j \leq 0\\
&t_{iw} \leq b_{iw}\\
&t_{wj} \leq b_{wj}\\
&-t_{iw} \leq -a_{iw}\\
&-t_{wj} \leq -a_{wj}\\
&\bullet\\
&\bullet\\
&t_f - t_0 \leq \lambda
\end{aligned}$$

Teorema:

Poiché $h_{iw} \geq h_{wj}$, per le ipotesi di convessità sulla funzione tempo-costo, all'ottimo $t_{wj} > 0$ solo se $t_{iw} = b_{iw}$.

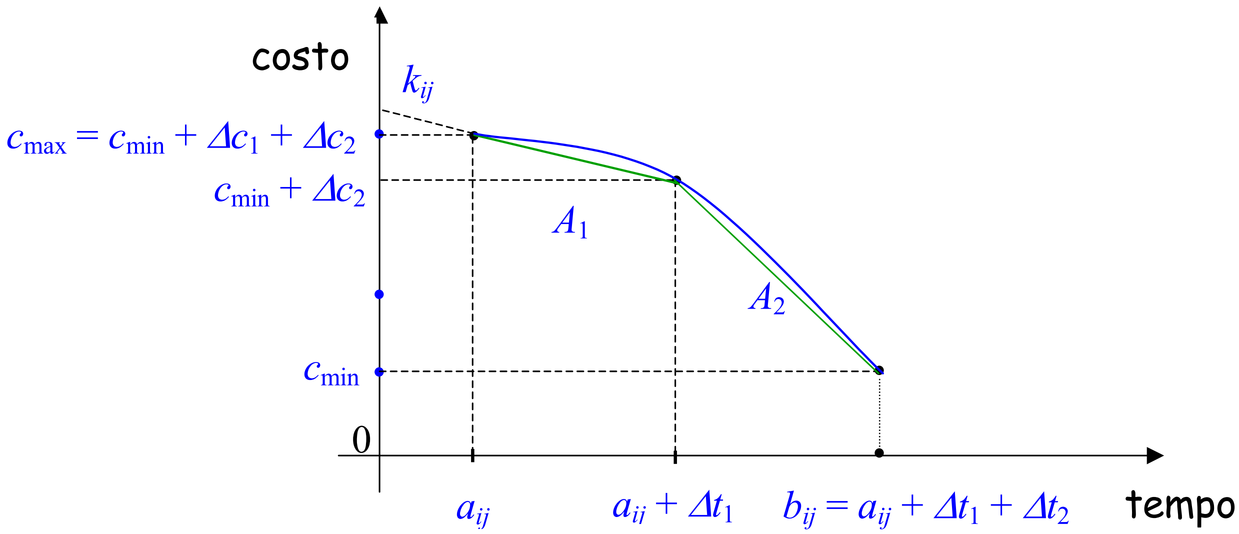
Project Scheduling: CPM least cost scheduling

10. Costi non lineari *(continua)*

- Questo approccio alle non-linearità può essere generalizzato, con qualche complicazione, al caso di relazioni tempo-costo del tutto generali purchè siano *continue e non crescenti*.

- La complicazione è dovuta al fatto che in generale non vale il precedente teorema.

- In particolare, supponiamo che l'attività A presenti una *relazione tempo-costo* continua non crescente e concava, come in figura.



Project Scheduling: CPM least cost scheduling

10. Costi non lineari *(continua)*

- La formulazione di pag. 39 non è più sufficiente a rappresentare correttamente il problema. Infatti non valendo più il precedente teorema, dobbiamo stavolta imporre esplicitamente la condizione per cui:

  se $t_{iw} < b_{iw}$ allora $t_{wj} = a_{wj} = 0$

- Dovendo aggiungere tale condizione al problema la formulazione diventa lineare intera mista per forzare opportunamente i valori delle t_{iw} e t_{wj}. Si introduce la variabile binaria ξ_{iw} che vale 1 se $t_{iw} = b_{iw}$ e 0 altrimenti.

$$
\begin{aligned}
\max \quad & z(\lambda) = \cdots + h_{iw}\, t_{iw} + h_{wj}\, t_{wj} + \cdots \\
\text{s.t.} \quad & \vdots \\
& t_i + t_{iw} - t_w \leq 0 \\
& t_w + t_{wj} - t_j \leq 0 \\
& t_{iw} \leq b_{iw} \\
& t_{wj} \leq b_{wj} \\
& -t_{iw} \leq -a_{iw} \\
& -t_{wj} \leq -a_{wj} \\
& b_{iw}\, \xi_{iw} - t_{iw} \leq 0 \\
& t_{wj} - b_{wj}\, \xi_{iw} \leq a_{wj} \\
& \vdots \\
& t_f - t_0 \leq \lambda \\
& \xi_{iw} \in \{0, 1\}
\end{aligned}
$$