Un progetto di ricerca e sviluppo di una società si compone di 12 (principali) attività con precedenze, durate normali b_{ij} (in giorni), costi diretti c_{ij} (in dollari) delle attività alla loro durata normale, durate crash a_{ij} (in giorni), ulteriori costi h_{ij} (in dollari) per ogni giorno in meno rispetto alla durata normale riportati nella tabella. Occorre inoltre considerare un costo indiretto di gestione del progetto pari a 100 $ al giorno.

ID	prec	b_{ij}	c_{ij}	a_{ij}	h_{ij}
A	-	30	1000	20	90
B	A	20	300	15	40
C	B,E	30	4000	20	200
D	A	10	300	6	50
E	A	14	1500	10	200
F	A	30	1200	25	150
G	D,E	20	900	12	80
H	D,E	80	1600	50	70
I	F,E	x	6000	x-5	240
J	G,I	30	12000	25	600
K	H,J	20	1500	15	400
L	C,K	5	2000	5	0

1) Disegnare la rete di progetto con le attività sugli archi, e la rete di progetto con le attività sui nodi.

2) L'attività I è composta da 5 sotto-attività:
T1, T2, T3, T4, T5 con le seguenti precedenze: T1 < T2; T2 < T3, T4, T5.

T1 richiede un consulente per 1 giorno;
T2 richiede un esperto per 3 giorni;
T3 richiede un esperto per 2 giorni;
T4 richiede un esperto e un consulente per 2 giorni.
T5 richiede un consulente per 4 giorni.

Le risorse umane a disposizione sono: 1 consulente, 1 esperto.

Si vuole determinare la durata minima x (all'ottimo, certificandola con un lower bound) dell'attività I in tale situazione. Risolvere il problema impiegando un approccio con modello su grafo. Descrivere l'approccio usato, e disegnare la schedula ottima delle sotto-attività.

3) Determinare il costo minimo derivante dall’esecuzione del progetto con le attività alle loro durate normali, descrivendo brevemente l’algoritmo impiegato e la sua complessità computazionale.

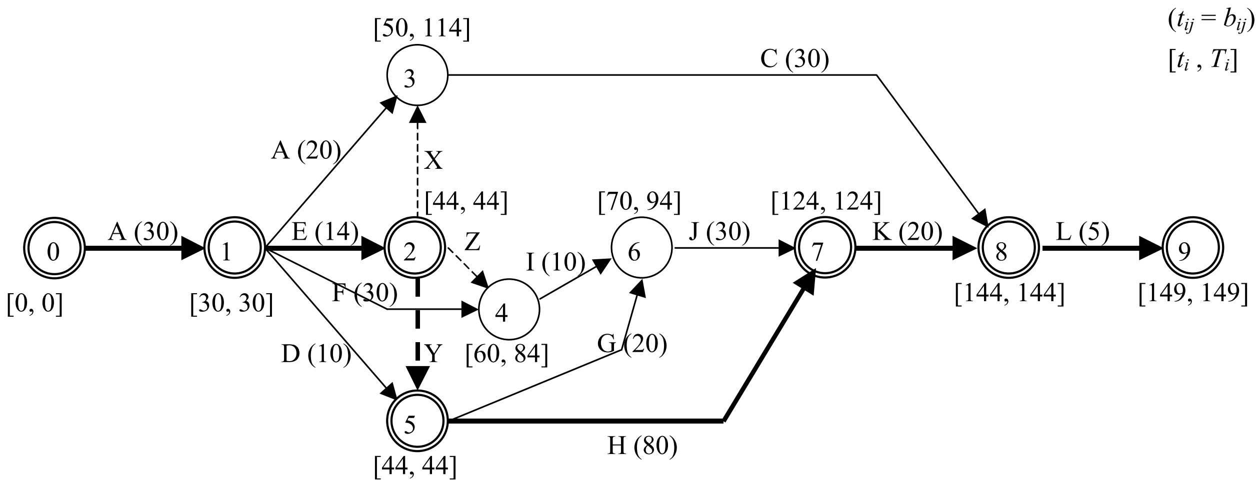
4) Indicare il(i) percorso(i) critico(i).

5) Si supponga che il progetto debba terminare entro 145 giorni (deadline), e che sia possibile ridurre la durata di tutte le attività al più fino alla loro durata crash; si vuole determinare il tempo totale richiesto per l’esecuzione del progetto in modo da rispettare la deadline e minimizzare i costi; riportare infine anche il costo ottimo.

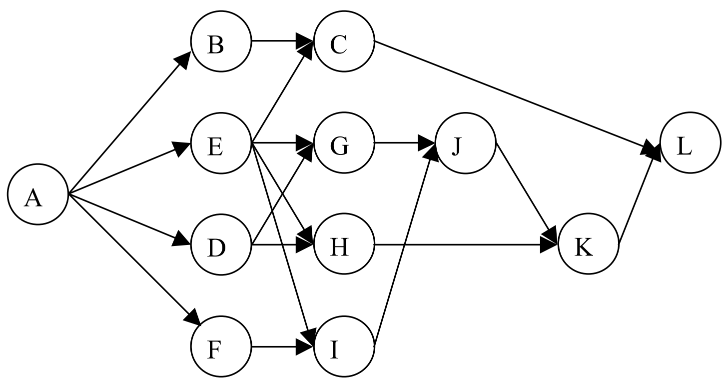
Soluzione

1)

Grafo con attività sugli archi (il disegno riporta anche le durate normali delle attività e, con queste durate, i tempi minimi e massimi di raggiungimento dei nodi, i nodi critici, le attività critiche e il percorso critico calcolati ai punti 3 e 4):



Grafo con attività sui nodi:

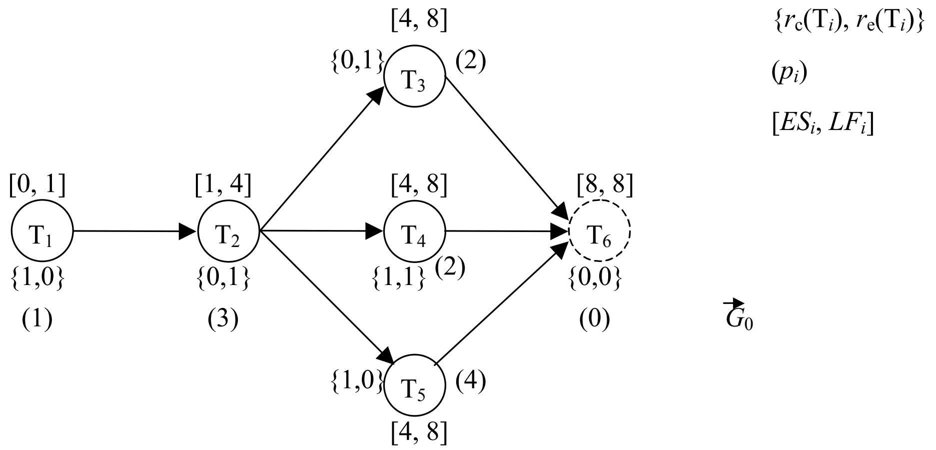


2)

Problema RCPSP. Si tratta di un problema NP-hard in senso forte.
Valuto lower bound sul makespan ottimo.

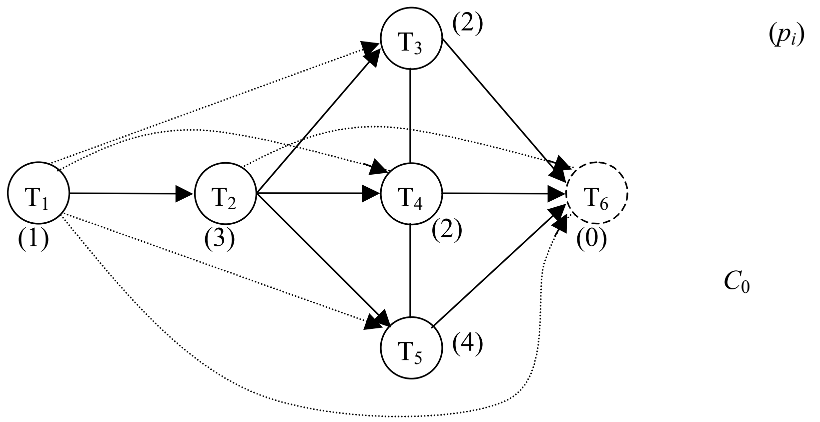
LB1) Si ottiene rilassando i vincoli sulle risorse, cioè supponendo le risorse sufficienti in quantità tale da permettere alle coppie di task indipendenti (senza relazione di precedenza) di poter essere eseguiti simultaneamente.

LB1 è pari alla lunghezza del percorso critico del grafo delle precedenze $\vec{G}_0$.



Applicando l'analisi dei tempi si ottiene LB1 = 8. Contestualmente si individuano minimi istanti di partenza (*ES*) e massimi istanti di fine (*LF*) delle attività, sempre in base alle ipotesi per il calcolo di LB1.

LB2) Si ottiene valutando il peso $\omega_p(C_0)$ della clique di peso massimo sul grafo dei conflitti C_0 pesato sui nodi con le durate dei task. Il grafo C_0 si ottiene chiudendo transitivamente il grafo delle precedenze G_0 e aggiungendo archi (non orientati) tra coppie di task T_i e T_j (non in relazione di precedenza) che non possono essere eseguiti simultaneamente perché esiste una risorsa R_h per cui $r_h(T_i) + r_h(T_j) > r_h$. Nel nostro caso occorre inserire due archi (T3, T4) e (T4, T5) in quanto $r_e(T_3) + r_e(T_4) = 2 > 1$ e $r_c(T_4) + r_c(T_5) = 2 > 1$. Il grafo dei conflitti è quindi:

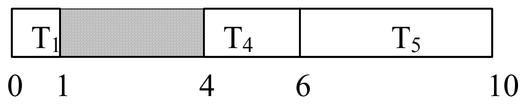


e la clique di peso massimo è $\{T_1, T_2, T_4, T_5, T_6\}$ di peso $\omega_p(C_0) = 10$. Quindi LB2 = 10.

LB3) Si ottiene rilassando tutti i vincoli sulle risorse tranne che per una risorsa. Si calcola il minimo tempo addizionale richiesto δ_s per schedulare i task sulla risorsa limitata *s*. Si ripete questo per ogni tipo *s* di risorsa. LB3 è pari a LB1 più il massimo tempo addizionale richiesto $\delta_{max} = \max_s\{\delta_s\}$. La risorsa *s*' per cui $\delta_{s'} = \delta_{max}$ è detta *bottleneck*.
Calcolare δ_s equivale a risolvere il problema ausiliario $1|r_i, \text{prec}|L_{max}$, sui task T_i che richiedono la risorsa *s*, con $r_i = ES_i$ e con $d_i = LF_i$.

Calcolo δ_c:

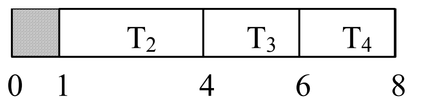
	T1	T4	T5
p_i	1	2	4
r_i	0	4	4
d_i	1	8	8



$\delta_c = L^*_{max} = 2$

Calcolo δ_e:

	T2	T3	T4
p_i	3	2	2
r_i	1	4	4
d_i	4	8	8

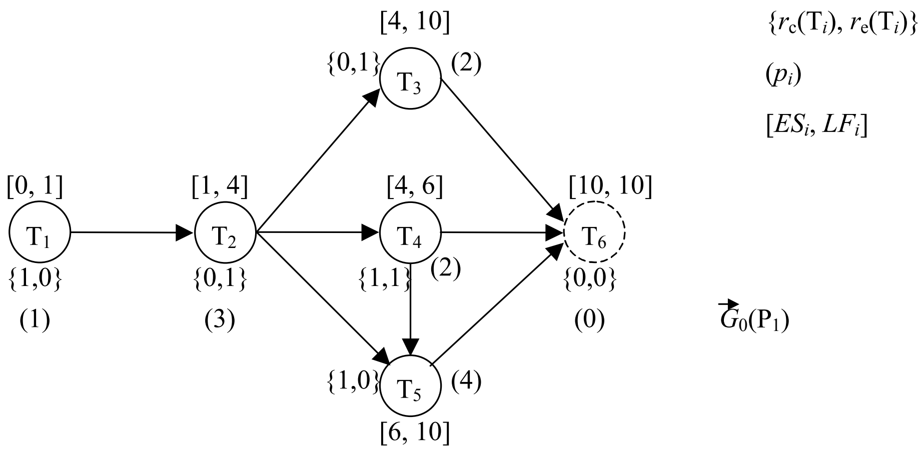


$\delta_e = L^*_{max} = 0$

LB3 = LB1 + δ_{max} = 8 + 2 = 10.

Determino soluzione ammissibile con euristica *shifting bottleneck*:
Ad ogni iterazione si determina la risorsa bottleneck tra le risorse non ancora schedulate e si sequenziano i task sulla risorsa bottleneck secondo la schedula che comporta il minimo tempo addizionale rispetto alla soluzione parziale ottenuta all'iterazione precedente.

1° iter) La ricerca della risorsa *bottleneck* tra la risorsa "consulente" e la risorsa "esperto" consiste nel ripetere il calcolo delle soluzioni dei due problemi ausiliari esaminati per il calcolo di LB3. Pertanto la risorsa bottleneck è la risorsa *consulente* e i task si sequenziano in base alla sequenza (T1, T4, T5) calcolata prima. Ciò comporta aggiungere al grafo $\vec{G}_0$ l'arco (T4 → T5). Si ottiene il grafo $\vec{G}_0(P_1)$.

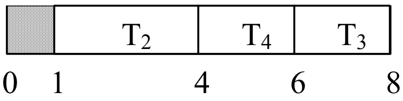


N.B.: Il grafo $\vec{G}_0(P_1)$ non è ammissibile (la sua chiusura transitiva contiene un insieme indipendente massimale di peso 2 pesando i nodi del grafo con $r_e(T_i)$ e quindi superiore a $r_e = 1$). Pertanto, occorre proseguire.

2° iter) La ricerca della risorsa *bottleneck* consiste nell'applicare la ricerca alla sola risorsa rimasta (*esperto*). Calcoliamo il minimo tempo addizionale richiesto δ_e.

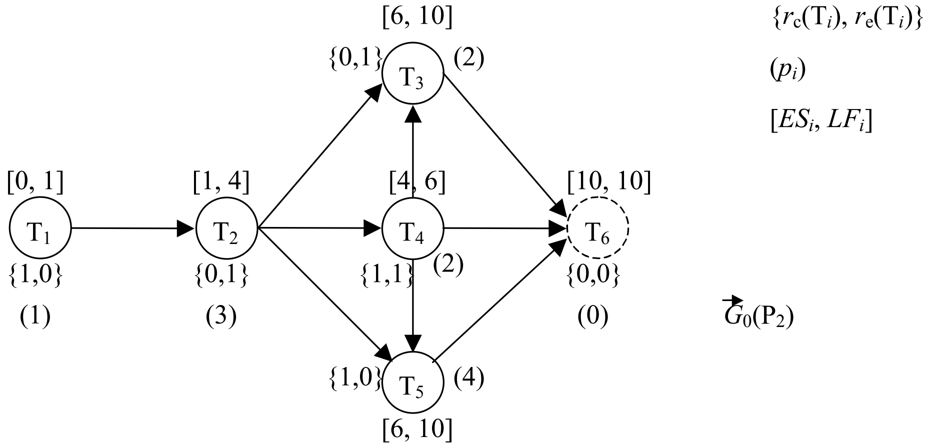
Calcolo δ_e:

	T2	T3	T4
p_i	3	2	2
r_i	1	4	4
d_i	4	10	6



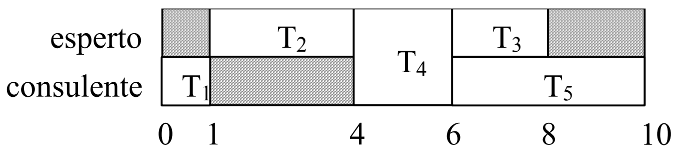
$\delta_e = L^*_{max} = 0$

Schedulare i task, secondo la sequenza (T2,T4,T3), sulla risorsa esperto non comporta ulteriore tempo addizionale. Pertanto l'aggiunta dell'arco (T4 $\rightarrow$ T3) al grafo $\vec{G}_0(P_1)$, non comporta aumento della lunghezza del percorso critico. Infatti, sia $\vec{G}_0(P_2)$ il grafo aggiunto ottenuto



Il grafo $\vec{G}_0(P_2)$ è ammissibile, nel senso che definisce una schedula ammissibile. Infatti, tutte le risorse sono assegnabili ai task in modo ammissibile, come si può anche verificare calcolando per ogni risorsa s il peso r_s^{max} dell'insieme indipendente di peso massimo della chiusura transitiva del grafo $\vec{G}_0(P_2)$ con i nodi pesati in base alla richiesta della risorsa $r_s(T_i)$ e verificando che $r_s^{max} \leq \bar{r}_s$.

La schedula ES associata a $\vec{G}_0(P_2)$ è:



Siccome $C_{max} = 10 = LB3$, la soluzione è ottima.

3)

Minimizzare i costi con le attività a durata normale significa minimizzare $c = \sum c_{ij} + 100\ t_9$, quindi minimizzare t_9, cioè determinare la minima durata del progetto con le attività a durata normale.

Occorre effettuare l'analisi dei tempi CPM con la procedura in avanti, che seguendo l'ordinamento topologico del grafo $G=(V,A)$ delle precedenze con attività sugli archi, permette di determinare i tempi minimi di raggiungimento t_j dei nodi, in cui $t_j = \max_{(i,j)\in A}\{t_i + t_{ij}\}$, avendo posto $t_0 = 0$. I valori sono riportati sul primo grafo disegnato al punto 1).
La complessità è $O(m)$ con $m = |A|$.
Dall'analisi dei tempi si ottiene che $t_9 = 149$, che è pertanto la minima durata del progetto con le attività a durata normale. Il costo ottimo in queste condizioni è $c^* = 32300 + 14900 = 47200$\$.

4)

Per determinare il(i) percorso(i) critico(i), occorre determinare le attività critiche. Queste sono definite come le attività (i,j) a tempo di slittamento totale $(T_j - E_{ij}) = 0$, dove $E_{ij} = t_i + t_{ij}$ è il tempo di completamento minimo dell'attività (i,j) e T_j è il tempo di raggiungimento massimo del nodo j corrispondente al tempo massimo di completamento dell'attività (i,j). Il tempo massimo di raggiungimento T_j di j è $T_j = \min_{(j,k)\in A}\{T_k - t_{jk}\}$, avendo posto $T_9 = t_9 = 149$.
C'è un solo percorso critico (A,E,Y,H,K,L), disegnato in grassetto sul primo grafo disegnato al punto 1).

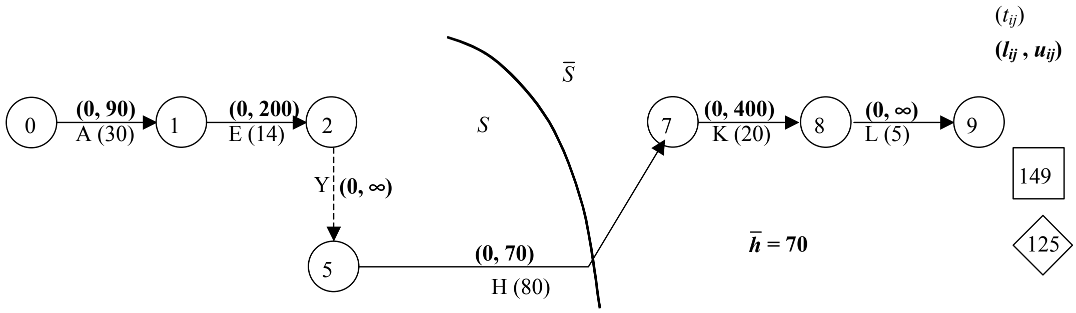
5)

Per determinare il costo minimo del progetto occorre ridurre la durata delle attività critiche in modo che $t_9 \le 145$ e che sia minima la variazione di costo rispetto al caso valutato al punto 3).
Si applica il *CPM least cost scheduling* che, a partire dalla situazione in cui tutte le attività sono a durata normale, iterativamente seleziona nella sottorete critica le attività che permettono di ridurre la durata del progetto al minimo aumento unitario $\overline{h}$ del costo diretto e riduce, modificando la durata delle attività selezionate, la durata del progetto per una massima entità di riduzione Δt.
In particolare:
- $\overline{h}$ è il taglio di capacità minima sulla sottorete critica in cui gli archi (i, j) sono capacitati secondo il seguente schema: $(0, h_{ij})$, (h_{ij}, h_{ij}), (h_{ij}, ∞), rispettivamente se l'arco (i, j) rappresenta un'attività a durata normale, intermedia o crash.
- $\Delta t = \min\{\Delta 1, \Delta 2\}$, dove $\Delta 1$ è il margine di riduzione della durata per le attività associate agli archi diretti del taglio (aumento della durata per le attività associate agli archi inversi del taglio) e $\Delta 2$ è la differenza tra la lunghezza del percorso critico e la lunghezza del percorso non critico più lungo che non contiene gli archi del taglio.

Il procedimento si arresta quando $t_9 \le 145$ e il rapporto $\Delta c/(-\Delta t) \ge 0$.

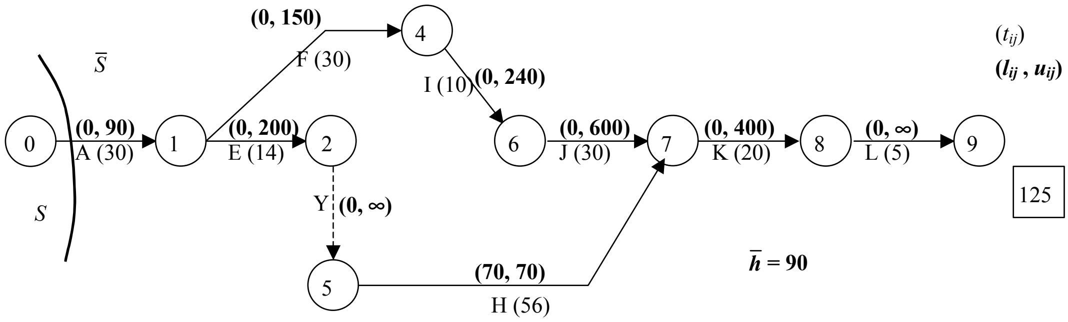
1° Iter)

Si effettua l'analisi dei tempi con le attività a durata normale. Si individua la sottorete critica. Si determina il taglio $(S, \overline{S})$ di capacità minima $\overline{h}$ sulla rete capacitata.



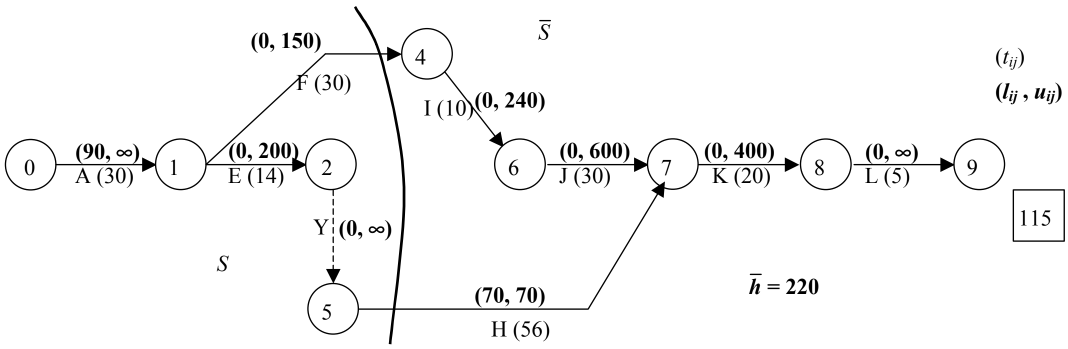
In base al taglio di capacità minima si ha che $\Delta c/(-\Delta t) = \bar{h} - 100 = 70 - 100 = -30$.
$\Delta 1 = t_{57} - a_{57} = 80 - 50 = 30$. $\Delta 2 = 149 - 125 = 24$, dove 125 è la lunghezza del percorso non critico più lungo che non contiene (5,6): cioè, il percorso (0,1,4,6,7,8,9).
Quindi, $\Delta t = \min\{30, 24\} = 24$ giorni.
Pertanto riducendo H di $\Delta t = 24$ giorni il progetto durerà 125 giorni con una variazione di costo totale $\Delta c = 24\ (-30) = -720\$$. Si noti che la durata del progetto in queste condizioni è ammissibile.

2° Iter)
Si effettua l'analisi dei tempi con le attività a durata stabilita al termine della iterazione precedente. Si individua la sottorete critica. Si determina il taglio $(S, \bar{S})$ di capacità minima $\bar{h}$ sulla rete capacitata.



In base al taglio di capacità minima si ha che $\Delta c/(-\Delta t) = \bar{h} - 100 = 90 - 100 = -10$. E' quindi conveniente ridurre la durata del progetto ulteriormente.
$\Delta 1 = t_{01} - a_{01} = 30 - 20 = 10$. $\Delta 2$ = - (tutti i percorsi contengono l'arco (0, 1)).
Quindi, $\Delta t = \min\{10, -\} = 10$ giorni.
Pertanto riducendo A di $\Delta t = 10$ giorni il progetto durerà 115 giorni con una variazione di costo totale $\Delta c = 10\ (-10) = -100\$$.

3° Iter)
Si effettua l'analisi dei tempi con le attività a durata stabilita al termine della iterazione precedente. Si individua la sottorete critica. Si determina il taglio $(S, \bar{S})$ di capacità minima $\bar{h}$ sulla rete capacitata.



In base al taglio di capacità minima si ha che $\Delta c/(-\Delta t) = \overline{h} - 100 = 220 - 100 = 120$. Non conviene quindi ridurre ulteriormente la durata del progetto. STOP.
In conclusione, riducendo la durata dell'attività H di 24 giorni e dell'attività A di 10 giorni, il progetto durerà 115 giorni con costo totale pari a 47200 + (-720) + (-100) = 46380$.