



Unione europea  
Fondo sociale europeo



**MINISTERO DEL LAVORO  
E DELLE POLITICHE SOCIALI**

Direzione Generale per le Politiche  
per l'Orientamento e la Formazione



REGIONE DEL VENETO

# Schedulazione delle attività di un progetto in presenza di multi-calendari

**Maria Silvia Pini**

Resp. accademico: Prof.ssa Francesca Rossi

Università di Padova



**Attività FSE DGR 1102/2010**

**La gestione dell'informazione nell'azienda**

Schedulazione intelligente di attività in presenza di risorse limitate e  
matching stabile ed efficiente tra domanda e offerta

# Schedulazione di attivita' - 1



- Assegnamento di **risorse** alle **attivit **  
**sull'asse temporale**
- **Risorse**
  - ▣ Macchine
  - ▣ Personale
  - ▣ Denaro
- **Attivit **
  - ▣ Fasi di un progetto
  - ▣ Servizi
  - ▣ Lezioni

# Schedulazione di attività - 2



- La schedulazione delle attività di un progetto determina **per ogni attività'**
  - **quando deve iniziare e quando deve finire,** considerando:
    - La sua durata
    - Le attività che la precedono
    - Le relazioni con i predecessori
    - La disponibilità delle risorse
    - La data di consegna del progetto

# Schedulazione di attività' - 3

## □ **Vincoli**

### □ **restrizioni temporali**

- ordine tra attività
- date di scadenza

### □ **capacità delle risorse**

- risorse consumabili e rinnovabili

## □ **Criterio di ottimizzazione**

### □ **makespan (lunghezza dello schedule)**

### □ **bilanciamento delle risorse**

### □ **ritardo sui tempi di consegna**

### □ **costo dell'assegnamento delle risorse**

# Schedulazione di attività' - 4



- **Variabili decisionali**

- Istante iniziale delle attività
- Istante finale delle attività (oppure durata se variabile)
- Risorse

- **Tipi di attività**

- interval activity: non può essere interrotta
- **breakable activity**: può essere interrotta da breaks
- preemptable activity: può essere interrotta da altre attività

# Esempio di schedulazione

- **6 attività**: ogni attività descritta da un predicato
  - **task**(**NAME**,**DURATION**,**LISTofPRECEDINGTASKS**,**MACHINE**)
    - task(j1,3,[],m1)
    - task(j2,8,[],m1)
    - task(j3,8,[j4,j5],m1)
    - task(j4,6,[],m2)
    - task(j5,3,[j1],m2)
    - task(j6,4,[j1],m2)
  - Le macchine m1 e m2 sono unarie (capacità 1).
  - Richiesto **massimo tempo di fine schedule**

# Breakable activities - 1



- In progetti reali dobbiamo considerare le **interruzioni** in cui **lavoratori e/o risorse** non sono disponibili
  - Week-end
  - Vacanze
  - Orario part-time
  - Si alternano **intervalli di lavoro** con **periodi di interruzione** (break)
- **Risorse diverse** possono avere **calendari diversi**

# Breakable activities - 2

- Data un'**attività**  $i$  otteniamo la corrispondente **attività' di calendario**  $b_i$  settando:
  - **$b_i(t) = 0$** , se c'è una **risorsa** usata dall'attività  $i$  che **non è disponibile** al tempo  $t$  a causa di un break
  - **$b_i(t) = 1$** , se **tutte le risorse** usate dall'attività  $i$  sono **disponibili** al tempo  $t$
  
- Presenteremo un **algoritmo di schedulazione** efficiente in questi scenari [Franck et al. 2001]
  - Tempi di inizio e di completamento delle attività'  
→ **il più presto possibile**



# Assunzioni dell'algoritmo

---

- Assunzioni accettate anche nella pratica
  - ▣ **Ogni attività** non può essere **interrotta** all'interno di un intervallo di lavoro, ma **solo all'inizio di un break**
  - ▣ **Ogni attività** deve **ripartire** nel primo istante di tempo del prossimo intervallo di lavoro
  - ▣ La **minima lunghezza** di un intervallo di **lavoro** tra due periodi di interruzione successivi deve essere almeno uguale ad **un'unità di tempo**
- In questa presentazione assumeremo inoltre che le risorse hanno una **capacità non limitata**

# Calendari con break

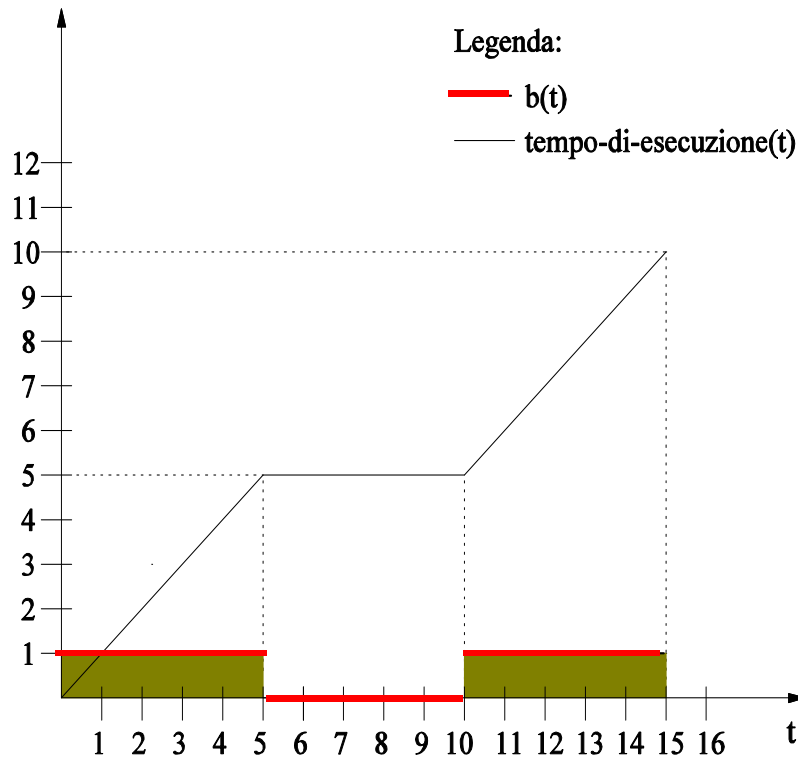
- Un **break-calendar** e' una **funzione**  $b: \mathbb{R}_{\geq 0} \rightarrow \{0,1\}$  tale che
  - $b(t) = 1$  se  $t$  appartiene a un periodo lavorativo
  - $b(t) = 0$  se  $t$  appartiene a un periodo non lavorativo
- Data una funzione di calendario  $b \rightarrow$  **il tempo totale di esecuzione** nell'intervallo  $[\alpha, \beta]$  e' dato da

$$\int_{\alpha}^{\beta} b(s) ds$$

Coincide con **l'area sottostante al grafico della funzione  $b$**  nell'intervallo che va **da  $\alpha$  a  $\beta$**

# Esempio - 1

Il calendario  $b$  e il corrispondente tempo totale di esecuzione

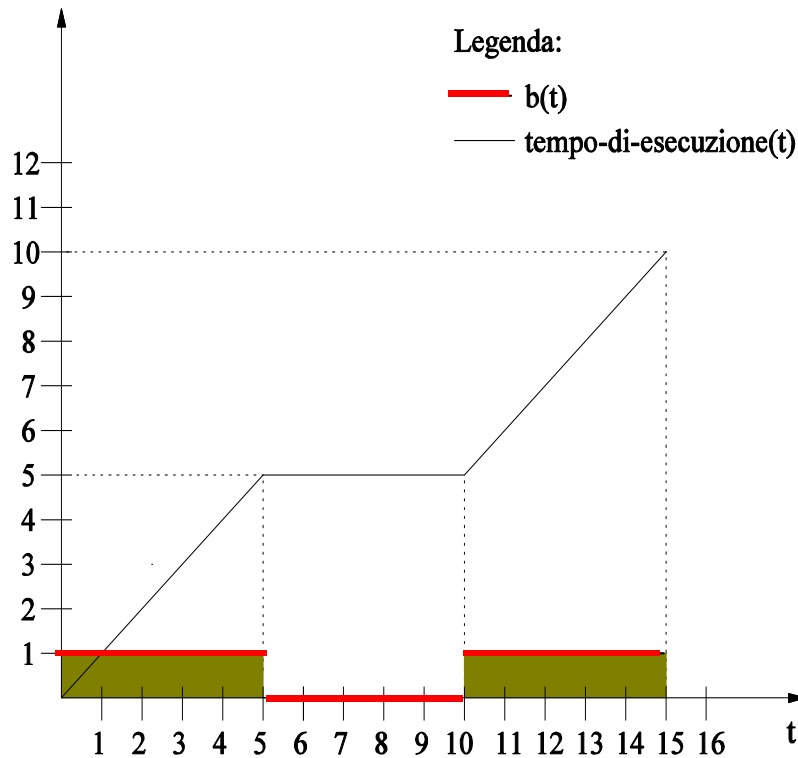


Intervalli di lavoro e intervalli di interruzione

- $[0,5[$  e  $[10,15[$  → intervalli di lavoro di una certa attività A1
- $[5,10[$  → break

# Esempio - 2

Il calendario  $b$  e il corrispondente tempo totale di esecuzione



Tempo di esecuzione

- Il tempo di esecuzione  $e'$ 
  - $t$  in  $[0, 5[$
  - 5 in  $[5, 10[$
  - $t-5$  in  $[10, 15[$
- Il tempo **totale** di esecuzione da  $t=0$  a  $t=15$  che è dato da

$$\int_0^{15} b(s) ds = 10$$

# L'algoritmo - 1

---

- Si basa sulle **assunzioni** fatte prima
  - **Ogni attività**' non puo' essere **interrotta** all'interno di un intervallo di lavoro
  - **Ogni attività**' deve **ripartire** nel primo istante di tempo del prossimo intervallo di lavoro
  - La **minima lunghezza** di un intervallo di **lavoro** tra due periodi di interruzione successivi deve essere almeno uguale ad **un'unita' di tempo**

# L'algoritmo - 2

- Si basa sul fatto che
  - ▣ data una funzione di calendario  $b \rightarrow$  **il tempo totale di esecuzione** nell'intervallo  $[\alpha, \beta]$  e' dato da 
$$\int_{\alpha}^{\beta} b(s) ds$$
cioe' dall'**area sottostante il grafico della funzione  $b$**  nell'intervallo da  $\alpha$  a  $\beta$
- Trova la **schedulazione ammissibile fatta al piu' presto**

# Descrizione dell'algoritmo - 1

## □ Input

- l'insieme  $V$  delle attività  $0, 1, \dots, n, n+1$ 
  - Le attività  $0$  e  $n+1$  sono attività fittizie che rappresentano l'inizio e la fine del progetto
- le funzioni di calendario  $b_0, b_1, \dots, b_n, b_{n+1}$  delle varie attività
- I pesi  $d_{ij}$  degli archi diretti tra le attività  $i$  e  $j$ , per ogni coppia di attività  $i$  e  $j$ 
  - Indicano che l'attività  $j$  può essere iniziata dopo almeno  $d_{ij}$  unità di tempo dall'inizio dell'attività  $i$

## □ Output

- vettore  $\mathbf{ES} = (\mathbf{ES}_0, \dots, \mathbf{ES}_{n+1})$  contenente i tempi di inizio delle attività
  - nel posto  $0$  del vettore  $\mathbf{ES}$  c'è il tempo di inizio del progetto
  - nel posto  $1$  del vettore  $\mathbf{ES}$  c'è il tempo di inizio dell'attività  $1$
  - ...
  - nel posto  $n+1$  del vettore  $\mathbf{ES}$  c'è il tempo di fine del progetto

# Descrizione dell'algoritmo - 2

- Inizialmente **ES**  $\leftarrow (0, -\infty, \dots, -\infty)$
- Poi le attività sono **ritardate** finché tutti i vincoli di calendario sono soddisfatti
- Consideriamo poi una **coda Q** che contiene le attività **1, 2, ..., n, n+1**, in cui è stato determinato un **tempo di inizio anticipato**
- In ogni iterazione, eliminiamo un attività *i* dalla coda Q



# Descrizione dell'algoritmo - 3

- Prima verifichiamo se **il tempo di inizio ESi rispetta oppure no il calendario bi**
- Per far questo, calcoliamo il primo istante di tempo  **$t^* \geq ESi$**  per cui l'intervallo  $[t^*; t^*+1[$  non contiene interruzioni
  - ▣ Se  **$t^* = \infty$**   $\rightarrow$  nessuna soluzione e' ammissibile e quindi l'algoritmo termina
  - ▣ Se  **$t^* > ESi$**   $\rightarrow$  ritardiamo il tempo di inizio dell'attivit  i fino al tempo  $t^*$

# Descrizione dell'algoritmo - 4

- Poi **controlliamo i vincoli temporali per tutti i successori diretti j** dell'attività i
- Per far questo calcoliamo il primo istante di tempo
$$t^* = \min\{t \geq \max\{0, ES_j\} \mid \int_{ES_i}^t b_{ij}(s) ds \geq dij\}$$

dell'attività j dato il tempo di inizio  $ES_i$  per l'attività i che la precede

- Se  $ES_j < t^*$ 
  - lo schedulazione ES non soddisfa i vincoli temporali, quindi settiamo  **$ES_j = t^*$**
- Se j non appartiene alla coda Q, lo inseriamo in Q
- Quest'ultima operazione la effettuiamo anche se  **$b_j(r) = 0$**  per qualche  $r$  in  $[t^*; t^* + 1 [$

# Funzionamento dell'algoritmo - 1

## Esempio

- 2 attività → A1 e A2
- A1 deve durare 10 ore
- A2 deve durare 5 ore
- A2 può essere effettuata solo dopo che A1 è finita
- Periodo lavorativo di A1 →  $[0,5[ \cup [10,15[$
- Periodo lavorativo di A2 →  $[7,12[ \cup [18,25[$

## Funzionamento dell'algoritmo

- Insieme delle attività  
 $V = \{A0, A1, A2, A3\}$
- $ES = (0, -\infty, -\infty, -\infty)$
- Coda Q contiene A1, A2, A3

# Funzionamento dell'algoritmo - 2

## Esempio

- 2 attività → A1 e A2
- A1 deve durare 10 ore
- A2 deve durare 5 ore
- A2 può essere effettuata solo dopo che A1 è finita
- Periodo lavorativo di A1 →  $[0,5[ \cup [10,15[$
- Periodo lavorativo di A2 →  $[7,12[ \cup [18,25[$

## Funzionamento dell'algoritmo

- **Iterazione 1:** elimino A1 dalla coda Q
  - ▣ Calcoliamo il primo istante di tempo  $t \geq ES1$  per cui l'intervallo  $[t; t+1[$  non contiene break, cioè  $t=0$
  - ▣ settiamo  **$ES1=0$**

# Funzionamento dell'algoritmo - 3

## Esempio

- 2 attività → A1 e A2
- A1 deve durare 10 ore
- A2 deve durare 5 ore
- A2 può essere effettuata solo dopo che A1 è finita
- Periodo lavorativo di A1 →  $[0,5[ \cup [10,15[$
- Periodo lavorativo di A2 →  $[7,12[ \cup [18,25[$

## Funzionamento dell'algoritmo

- Poi controlliamo i vincoli temporali per tutti i **successori diretti** dell'attività A1 (cioè A2 nell'esempio)
  - Per far questo calcoliamo il primo istante di tempo
$$t^* = \min \{t \geq \max \{0, ES2\} \mid \int_{ES1}^t b12(s) ds \geq 10\}$$
  - $t^* = 15$
  - quindi settiamo **ES2 = 15**
- L'attività A2 è già in Q, quindi non la devo inserire in Q

# Funzionamento dell'algoritmo - 4

## Esempio

- 2 attività → A1 e A2
- A1 deve durare 10 ore
- A2 deve durare 5 ore
- A2 può essere effettuata solo dopo che A1 è finita
- Periodo lavorativo di A1 →  $[0,5[ \cup [10,15[$
- Periodo lavorativo di A2 →  $[7,12[ \cup [18,25[$

## Funzionamento dell'algoritmo

- **Iterazione 2:** elimino A2 dalla coda Q
  - ▣ Calcoliamo il primo istante di tempo  $t \geq ES_2 = 15$  per cui l'intervallo  $[t; t+1[$  non contiene break, cioè  $t = 18$
  - ▣ settiamo  $ES_2 = 18$

# Funzionamento dell'algoritmo - 5

## Esempio

- 2 attività → A1 e A2
- A1 deve durare 10 ore
- A2 deve durare 5 ore
- A2 può essere effettuata solo dopo che A1 è finita
- Periodo lavorativo di A1 →  $[0,5[ \cup [10,15[$
- Periodo lavorativo di A2 →  $[7,12[ \cup [18,25[$

## Funzionamento dell'algoritmo

- Controlliamo i vincoli temporali per tutti i **successori diretti** dell'attività A2 (cioè l'attività di fine progetto A3 )
  - Per far questo calcoliamo il primo istante di tempo
$$t^* = \min \{ t \geq \max \{ 0, ES3 \} \mid \int_{ES2}^t b_{23}(s) ds \geq 5 \}$$
  - $t^* = 23$
  - settiamo **ES3 = 23**
- L'attività A3 è già in Q, quindi non la devo inserire in Q

# Funzionamento dell'algoritmo - 6

## Esempio

- 2 attività → A1 e A2
- A1 deve durare 10 ore
- A2 deve durare 5 ore
- A2 può essere effettuata solo dopo che A1 è finita
- Periodo lavorativo di A1 →  $[0,5[ \cup [10,15[$
- Periodo lavorativo di A2 →  $[7,12[ \cup [18,25[$

## Funzionamento dell'algoritmo

- **Iterazione 3:** elimino A3 dalla coda Q (che diventa vuota)
- A3 ha durata 0 e non ha successori, quindi nessuna altra attività viene inserita in Q
- Quindi l'algoritmo termina
- OUTPUT: vettore aggiornato dei tempi di inizio delle attività A0, A1, A2, A3 cioè
- **ES = (0,0,18,23)**



# Funzionamento dell'algoritmo - 7

## Esempio

- 2 attività → A1 e A2
- A1 deve durare 10 ore
- A2 deve durare 5 ore
- A2 può essere effettuata solo dopo che A1 è finita
- Periodo lavorativo di A1 →  $[0,5[ \cup [10,15[$
- Periodo lavorativo di A2 →  $[7,12[ \cup [18,25[$

## Funzionamento dell'algoritmo

- **ES = (0,0,18,23)**
  - ▣ Inizio del progetto: istante 0
  - ▣ Inizio dell'attività A1: istante 0
  - ▣ Inizio dell'attività A2: istante 18
  - ▣ Inizio dell'attività A3 (cioè fine del progetto): istante 23

# Conclusioni



- Abbiamo considerato il problema della schedulazione di attività in contesti in cui
  - ▣ le varie attività hanno **diversi calendari**
  - ▣ ci sono dei **vincoli temporali** tra le varie attività
- Abbiamo presentato un algoritmo che restituisce la **schedulazione ammissibile al più presto** delle attività che fa delle assunzioni che sono generalmente accettate nella pratica



Unione europea  
Fondo sociale europeo



**MINISTERO DEL LAVORO  
E DELLE POLITICHE SOCIALI**

Direzione Generale per le Politiche  
per l'Orientamento e la Formazione



REGIONE DEL VENETO

# Schedulazione delle attività di un progetto in presenza di multi-calendari

**Maria Silvia Pini**

Resp. accademico: Prof.ssa Francesca Rossi

Università di Padova



**Attività FSE DGR 1102/2010**

**La gestione dell'informazione nell'azienda**

Schedulazione intelligente di attività in presenza di risorse limitate e  
matching stabile ed efficiente tra domanda e offerta