



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 e di vincoli sulle risorse

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

Il problema

- Schedulazione delle attività di un progetto
 - ▣ **Vincoli temporali**
 - Alcune attività devono essere eseguite prima di altre
 - ▣ **Vincoli di calendario**
 - Ogni attività ha un proprio calendario
 - ▣ **Vincoli sulle risorse**
 - Ogni risorsa k ha una capacità limitata R_k
 - Processare l'attività i nell'intervallo $[S_i, C_i[$ richiede r_{ik} unità di risorsa
 - S_i è il tempo di inizio dell'attività i
 - C_i è il tempo di completamento dell'attività i
- **Scopo:** trovare una schedulazione feasible (dal punto di vista delle risorse e dal punto di vista dei calendari) che minimizza la durata del progetto

L'algoritmo - 1

- Per risolvere il nostro problema vedremo un **metodo che si basa sulle prioritá** [Franck et al. 2001]
- Sia $<$ un **ordinamento stretto** nell'insieme delle attivita'
 - Siano i e j due attivita'
 - $i < j$ se
 - $d_{ij} > 0$ oppure
 - $d_{ij} = 0$ e $d_{ji} < 0$
 - dove d_{ij} e' la lunghezza del piu' lungo cammino dal nodo i al nodo j nel grafo delle dipendenze tra le attivita'
- L'algoritmo schedula le attivita' una dopo l'altra

L'algoritmo - 2

- Sia C l'insieme delle attività già schedulate
- Sia C' l'insieme delle attività da schedulare
- Ad ogni iterazione
 - Un'attività "eligible" di C' viene schedulata al suo minimo tempo di inizio che è fattibile dal punto di vista delle risorse ($t^* = ESj^*$, a condizione che $t^* \leq LSj^*$)
 - Un'attività **eligible** j^* è un'attività che
 - non è ancora stata schedulata e
 - le cui attività $i < j^*$ nell'ordinamento sono già state schedulate
 - ESj^* = primo istante di inizio possibile per l'attività j^*
 - LSj^* = ultimo istante di inizio possibile per l'attività j^*

L'algoritmo - 3

- ▣ Se ci sono piu' attivita' eligible
 - j^* viene scelto secondo una regola di prioritá'
 - La miglior regola di prioritá' seleziona l'attivita' eligible con il **piu' piccolo tempo di inizio finale**
- ▣ In seguito vengono aggiornati i primi e gli ultimi istanti di inizio delle varie attivita' non ancora schedulate
 - $\forall i \in C'$, vengono aggiornati ES_i e LS_i
 - Per aggiornare
 - **ES_i** si usa l'algoritmo visto nella presentazione precedente (senza vincoli sulle risorse) **ALG.1 (schedulazione al piu' presto)**
 - **LS_i** si usa un algoritmo simile a quello visto nella presentazione precedente **ALG.2 (schedulazione al piu' tardi)**
- ▣ Poi **vengono aggiornate le capacita' delle risorse**

L'algoritmo - 4

- If $t^* > LS_j^*$
 - ▣ L'attività j^* non può essere schedulata senza violare un vincolo temporale con qualche attività i già schedulata
 - ▣ Facciamo un **passo di non-schedulazione**
 - **rimuoviamo** da C (l'insieme delle attività già schedulate) tutte le attività h che hanno tempo di inizio minore o uguale a quello di i , cioè $S_h \leq S_i$
 - **Aumentiamo** il primo tempo di inizio possibile per l'attività i , cioè ES_i , di $t^* - LS_j^*$

ALG. 1 (schedulazione al piu' presto)

- Per aggiornare **ESi** delle attivita' i non ancora schedulate si utilizza l'algoritmo che abbiamo presentato nella riunione precedente
 - Inizialmente **ES** $\leftarrow (0, -\infty, \dots, -\infty)$
 - Poi le attivita' sono **ritardate** finche' tutti i vincoli di calendario sono soddisfatti
 - Consideriamo poi una **coda Q** che contiene le attivita' **1, 2, ..., n, n+1**, in cui e' stato determinato un **tempo di inizio anticipato**
 - In ogni iterazione, eliminiamo un attivita' i dalla coda Q

ALG. 2 (schedulazione al piu' tardi)

- Per aggiornare LS_i delle attivita' i non ancora schedulate si utilizza un algoritmo simile al precedente
 - Sia d' la **data di consegna del progetto** o la massima durata del del progetto
 - Partiamo con $LS \leftarrow (0, \infty, \dots, \infty, d')$
 - Poi e attivita' sono **anticipate** finche' tutti i vincoli di calendario sono soddisfatti
 - Consideriamo una **coda Q** che contiene le attivita' 0 e $n+1$
 - In ogni iterazione
 - Prima **eliminiamo un attivita' j dalla coda Q** e settiamo $LS_j = t^*$ (ultimo tempo di inizio dell'attivita' j permesso dal calendario b_j)
 - Poi settiamo il potenziale ultimo tempo di inizio LS_i , $\forall i$ **predecessore di j**, con il piu' grande tempo t^* tale che

$$t^* \leq \min(d', LS_i) \quad \text{e} \quad \int_{t^*}^{LS_j} b_{ij}(s) ds \geq d_{ij}$$