

## Introduzione al Project Scheduling

- Da qui in avanti, faremo riferimento alla *gestione della tempistica* di un *progetto* e in particolare al processo di *determinazione della schedula di progetto*.
- Supporremo pertanto di aver effettuato il processo di pianificazione del contenuto del progetto e tramite lo strumento della WBS aver definito la lista delle attività del progetto.
- Supporremo inoltre di aver definito le relazioni di precedenza fra le attività e per le attività di aver stimato la durata e le risorse necessarie.
- In particolare, esamineremo alcune *tecniche* per la *produzione di schedule delle attività di un progetto*.
- Ci occuperemo pertanto delle problematiche del *project scheduling*
- Assumeremo quindi che il progetto è costituito da un *insieme di attività*, per le quali sono state definite le *relazioni di precedenza*, le *stime delle durate*, il *fabbisogno delle risorse*, un insieme di *risorse disponibili*, ed eventuali ulteriori aspetti specifici.

# Introduzione al Project Scheduling

## 1. Classificazione dei problemi

- E' possibile *classificare* i *problemi* di *project scheduling*, analogamente a quanto fatto per i problemi di scheduling su macchina, cioè in base a:
  - *Ambiente delle risorse*;
  - *Caratteristiche delle attività*;
  - *Obiettivo da perseguire*.

### Ambiente delle risorse

- Si caratterizza sulla base di:
  - *Tipologia delle risorse*: rinnovabili, non rinnovabili; doppiamente vincolate; ...
  - *Quantità delle risorse*: illimitata; limitata.
  - *Modalità di esecuzione delle attività*: single-mode; multi-mode.

# Introduzione al Project Scheduling

## 1. Classificazione dei problemi (continua)

### Caratteristiche delle attività

- Si caratterizza sulla base di:
  - *Tipo di relazione tra le attività*: chain; intree; outtree; tree; generiche; ...
  - *Tipo di precedenze*: finish-to-start; generalizzate.
  - *Vincoli temporali tra le attività*: time lag associati alle relazioni di precedenza.
  - *Natura delle durate*: deterministica; stocastica.
  - *Vincoli temporali sulle attività*: finestre temporali.
  - *Deadline sul completamento progetto*.

### Obiettivo da perseguire

- Possibili obiettivi: *minimizzazione del tempo di completamento del progetto; massimizzazione del valore attuale netto; livellamento uso delle risorse; minimizzazione dei costi totali; minimizzazione della lateness del progetto; minimizzazione del flow time totale.*

# Introduzione al Project Scheduling

## 2. Principali macro classi

- E' possibile suddividere i problemi di *project scheduling* in *due macro classi* principali:

1) *Project scheduling con risorse illimitate* (o senza risorse aggiuntive):  $PS_{\infty}|\beta|\gamma$

Tipicamente in tali problemi si chiede di:

- Minimizzare il tempo di completamento del progetto ( $\gamma = C_{\max}$ ) rispettando le relazioni di precedenza tra le attività ( $\beta = prec, \dots$ ).
- Minimizzare i costi delle attività con il vincolo di terminare il progetto entro una certa data, dove il costo di un'attività è una funzione non crescente della sua durata.

Sono problemi *trattabili* (computazionalmente).

2) *Project scheduling con risorse limitate*:  $PS|\beta|\gamma$

- Di solito si fa riferimento a risorse discrete.
- Si possono avere finestre temporali per le attività.
- Altri vincoli ...

Sono problemi in generale *NP-hard in senso forte*.

# Introduzione al Project Scheduling

## 3. Le tecniche reticolari

- Nella fase di scheduling vengono usate *diverse tecniche* di rappresentazione che possono essere classificate in *due tipologie* fondamentali:

(i) *Diagrammi di Gantt*: in questa tecnica le attività vengono rappresentate come "barre" per cui viene specificata la data di inizio e quella di fine.

- Questa tecnica, spesso usata per la sua *espressività, chiarezza e comprensibilità*, appare inadeguata in progetti complessi in quanto *non mostra tutte le interdipendenze tra le attività* ed è difficile da usare nella fase di rescheduling.

(ii) *Diagrammi reticolari*: queste tecniche usano una *rappresentazione del progetto su una rete* o grafo orientato secondo due distinte modalità: *attività sui nodi* o *attività sugli archi*.

- Nel primo caso i nodi rappresentano attività e gli archi relazioni di precedenza tra esse. Nel secondo caso gli archi rappresentano attività ed i nodi rappresentano eventi. In entrambi i casi le relazioni di precedenza sono rappresentate in modo esplicito.

# Introduzione al Project Scheduling

## 3. Le tecniche reticolari

(continua)

- I diagrammi reticolari sono alla base delle *tecniche reticolari* per il project scheduling con risorse illimitate, che si sono sviluppate a partire dal 1958 risultando particolarmente efficienti.
- Le più importanti e note sono:
  - (i) *Program Evaluation and Review Technique (PERT)*
  - (ii) *Critical Path Method (CPM): CPM time analysis; CPM time-cost tradeoff*
  - (iii) *Precedence Diagramming Method (PDM): GPR project network time-analysis*
- Il *PERT* fu sviluppato per pianificare la produzione del missile "Polaris" e portò al risparmio di due anni sui cinque preventivati per sviluppare il sistema.
- Il *PERT*, sviluppato per pianificare e controllare i tempi di esecuzione di un progetto attraverso un trattamento statistico delle incertezze associate ai tempi di completamento delle *attività* quando la loro *durata* è *stocastica*, è stato anche esteso alla pianificazione e controllo dei costi e della qualità del prodotto finale.

# Introduzione al Project Scheduling

## 3. Le tecniche reticolari (continua)

- Il **CPM**, sviluppato in campo industriale, si applica ai progetti per i quali le **durate** delle **attività** sono **deterministiche**.
- Il **CPM time-analysis** si occupa di effettuare l'analisi della tempistica delle attività con l'obiettivo di minimizzare il tempo d'esecuzione del progetto (senza vincoli sulle risorse,  $PS_{\infty} | prec | C_{max}$ ).
- Il **CPM time-cost tradeoff** prende in esame durate e costi delle attività e ha come obiettivo la minimizzazione dei costi di esecuzione (delle attività) di un progetto ( $MPS_{\infty} | prec, \bar{d} | \sum c_j(p_j)$ ).
- La **GPR project network time-analysis** effettua l'analisi della tempistica delle attività di un progetto con relazioni di precedenza di tipo generalizzato (GPR) con l'obiettivo di minimizzare il tempo d'esecuzione (senza vincoli sulle risorse,  $PS_{\infty} | GPR | C_{max}$ ).

### Osservazione

- Sia il **PERT** che il **CPM** e la sua generalizzazione sulle reti **GPR** utilizzano una rappresentazione del progetto su una rete secondo opportune regole che consentono al manager di **concentrarsi su** quelle **attività più critiche** ( $10 \div 20\%$  del totale).

# Introduzione al Project Scheduling

## 4. Grafo di progetto

- In generale, occorre innanzitutto:
  - (i) *Identificare* tutte le *attività* che devono essere completate prima che il progetto possa avere termine.
  - (ii) Per ogni attività occorre conoscere le *precedenze*, ossia quali attività devono essere completate prima di poter iniziare l'attività in esame.
- Sia quindi dato l'insieme delle attività insieme alle loro relazioni di precedenza che assumeremo di tipo finish-to-start (con time-lag minimo nullo).
- Il primo passo per applicare il *PERT* il *CPM* e il *PDM* consiste nel rappresentare le attività e le relazioni di precedenza su una *rete orientata*  $G = (N, A)$ .
- Per le ipotesi fatte sulla tipologia di precedenze la rete è *aciclica*.
- In particolare, la *rete* utilizzata nella applicazione del *PERT* del *CPM* può essere di tipo *AOA* (con attività sugli archi) o di tipo *AON* (con attività sui nodi). La *PDM* usa tipicamente la rappresentazione *AON*.

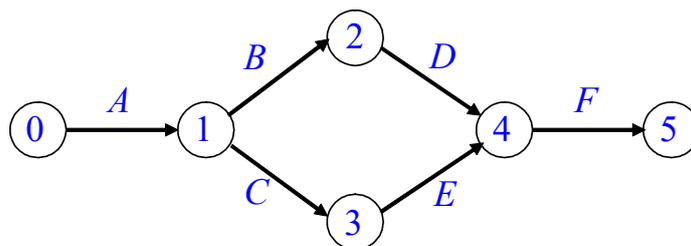
# Introduzione al Project Scheduling

## 4. Grafo di progetto

(continua)

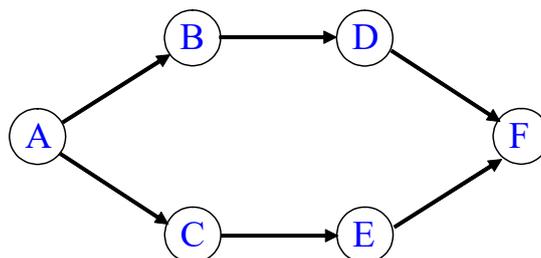
Rete **AOA**  $G = (N, A)$

- In tal caso:
  - Ogni **arco**  $(i, j) \in A$  corrisponde ad un'**attività**.
  - Ogni **nodo**  $i \in N$  rappresenta un **evento**.
    - Esempio: inizio progetto, fine progetto, completamento di un insieme di attività.



Rete **AON**  $G = (N, A)$

- In tal caso:
  - Ogni **nodo**  $i \in N$  rappresenta un'**attività**.
  - Ogni **arco**  $(i, j) \in A$  rappresenta una **relazione di precedenza** (nell'esempio di tipo standard) tra una coppia di attività.



# Introduzione al Project Scheduling

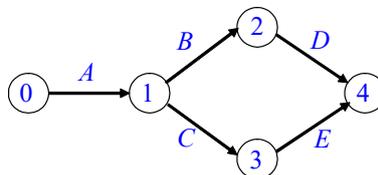
## 4.1. Grafo di progetto: rete AOA

- Facciamo uso della rete AOA  $G = (N, A)$ 
  - Nel *disegnare* la rete AOA di progetto occorre osservare le seguenti *regole*:
    - 1) Le *attività* sono rappresentate dagli *archi*.
    - 2) Gli *archi diretti verso un nodo* rappresentano *attività da completare prima* che abbiano *inizio attività* rappresentate da *archi che hanno origine in quel nodo*.
    - 3) *Lunghezza e forma archi non hanno significato*.
    - 4) *Due nodi non possono essere connessi da più di un arco*.
    - 5) L'*inizio del progetto* è *rappresentato* da un *nodo* contrassegnato con *zero* e tutti gli *altri* sono *numerati in modo che se esiste un arco diretto dal nodo  $i$  al nodo  $j$  risulti  $i < j$* .
    - 6) Rete con *un solo nodo iniziale, un solo nodo finale*

### Esempio

Attività:  $A, B, C, D, E$

Precedenze:  $A < B; A < C; B < D; C < E$



# Introduzione al Project Scheduling

## 4.1. Grafo di progetto: rete AOA (continua)

### Osservazione

- Le *prime tre regole* rispecchiano la *logica interna della rete* mentre le *altre* sono *conseguenza dei metodi di soluzione* che in generale richiedono l'uso di algoritmi di calcolo da implementare su calcolatore.

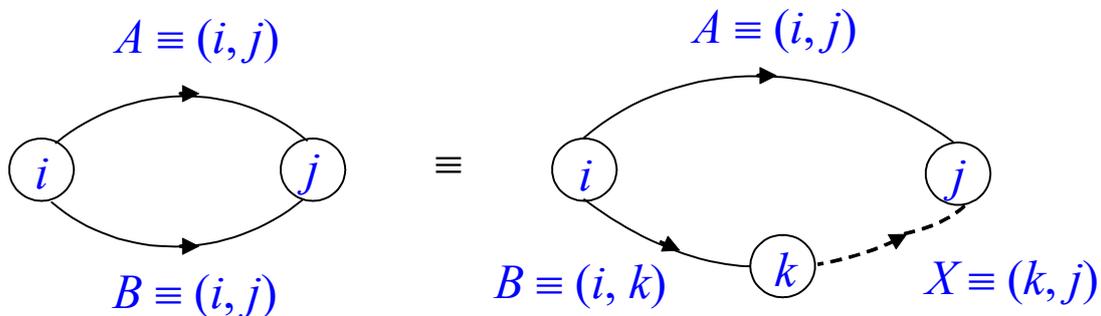
### Introduzione di attività fittizie

- Quando le regole precedenti per costruire il grafo rappresentativo del progetto non sono soddisfatte si ricorre a degli artifici.
- In particolare, *a volte è necessario introdurre attività fittizie che hanno durata nulla.*
- Chiaramente si deve introdurre il *minimo numero necessario* di tali attività per non complicare inutilmente la rete di progetto e gli algoritmi di calcolo.

# Introduzione al Project Scheduling

## 4.1 Grafo di progetto: rete AOA (continua)

- In particolare, *se due attività hanno gli stessi nodi iniziale e finale violerebbero la regola 4.*
- In tal caso *si può procedere introducendo una attività fittizia con tempo di esecuzione pari a zero* come nell'esempio seguente:

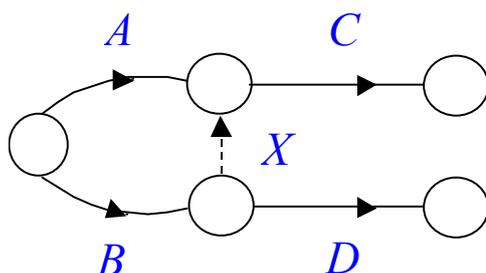


$X \equiv (k, j)$  *attività fittizia*

- Un altro caso in cui occorre introdurre un'attività fittizia è il seguente:

$A, B, C, D$  sono 4 *attività* tali che  $A, B, < C$  e  $B < D$ .

Dalla figura seguente si vede l'*esigenza di introdurre un'attività fittizia* per rappresentare tutte le precedenze.



$X$  *attività fittizia*

# Introduzione al Project Scheduling

## 4.2. Grafo di progetto: rete AON

- Nel *disegnare* la rete AON di progetto occorre osservare le seguenti *regole*:
  - 1) Le *attività* sono *rappresentate* dai *nodi*.
  - 2) Le *relazioni di precedenza* sono *rappresentate* dagli *archi*.
  - 3) *Lunghezza e forma archi non hanno significato*.
  - 4) Rete con *un solo nodo iniziale, un solo nodo finale*.
- Nel *disegnare* la rete AON non sono necessari archi o nodi fittizi, tranne che per garantire l'unicità della sorgente e del pozzo della rete.

### Esempio

Attività:  $A, B, C, D, E$

Precedenze:  $A < B; A < C; B < D; C < E$

