

# Introduzione alla teoria dello scheduling

## 1. Generalità

I problemi di scheduling riguardano l'*allocazione di risorse limitate ad attività nel tempo*.

In generale il processo di decisione associato richiede la *determinazione dell'ordine* in cui l'insieme di attività viene eseguito, il *tempo* in cui ogni attività viene eseguita perseguendo un certo obiettivo.

*Risorse, attività, e obiettivo* possono essere di diversa natura. In particolare:

1. Le *risorse* possono essere le macchine in un'officina, le piste in un aeroporto, gli equipaggi in un sistema di trasporto, le unità di calcolo in un centro di calcolo e così via.
2. Le *attività* possono essere le operazioni in un processo produttivo, i decolli e gli atterraggi in un aeroporto, le tratte di volo nei viaggi aerei, l'esecuzione dei programmi in un centro di calcolo e così via.
3. L'*obiettivo* può essere la minimizzazione del tempo di completamento dell'ultima attività, la minimizzazione del tempo medio di completamento.

# Introduzione alla teoria dello scheduling

## 1. Generalità

(continua)

Lo *scheduling* è pertanto un processo di decisione che si pone in *svariati contesti applicativi* quali i *sistemi manifatturieri, sistemi di elaborazione, sistemi di distribuzione e trasporti ed altri sistemi di servizio*.

Osservazione 1: con riferimento alla determinazione dell'ordine delle attività si usa spesso il termine *sequenziamento* invece di *scheduling* in quanto la definizione dell'ordine messo in relazione alla durata delle attività induce automaticamente anche la loro allocazione temporale.

Osservazione 2: poiché i problemi di sequenziamento vengono in realtà risolti, ci si potrebbe domandare quale è la novità del problema.

In passato questi problemi venivano risolti senza esaminare la bontà della soluzione ossia senza valutare le differenze di costo derivanti da una soluzione rispetto ad un'altra.

Nel caso di sistemi complessi l'ottimalità della soluzione diventa economicamente rilevante per cui i problemi di sequenziamento si risolvono ricorrendo alla teoria dello scheduling.

# Introduzione alla teoria dello scheduling

## 1. Generalità

(continua)

### Campi di applicazione

- *Shop scheduling*
  - Shop floor control (nei sistemi manifatturieri)
- *Schedulazione del personale*
  - Personale di un ospedale
  - Operai negli impianti di produzione
  - Crew scheduling
- *Scheduling di facility*
  - Centri espositivi
  - Sale operatorie
  - Aule didattiche
- *Scheduling di veicoli*
  - Spedizioni postali
  - Schedulazione di aerei, treni, autobus
- *Project scheduling*
- ...

# Introduzione alla teoria dello scheduling

## 1. Generalità

(continua)

### Esempi:

- Un'azienda produttrice di sacchetti di carta per cemento, cibo per cani, carbone, etc. La materia prima è costituita da rotoli di carta. Il processo produttivo consiste di tre stadi: stampa del logo sui rotoli, incollaggio dei lati del sacchetto, cucitura di una o due estremità del sacchetto. A ogni stadio, l'operazione è effettuata impiegando uno specifico operaio. Ogni ordine di produzione riporta il numero di sacchetti di un certo tipo che devono essere prodotti e consegnati al cliente entro una certa data. Il tempo di lavorazione è proporzionale al quantitativo di sacchetti ordinati. La consegna in ritardo rispetto alla data di consegna comporta un costo di penalità dipendente dall'entità del ritardo e dall'importanza dell'ordine (ovvero del cliente). Noti i tempi richiesti dalle varie operazioni per ciascun ordine, un obiettivo della pianificazione è quello di organizzare le lavorazioni in modo tale da minimizzare queste penali.
- Per il medesimo caso di prima, supponiamo che ciascuna fase di lavorazione avvenga presso una stazione di lavoro che va attrezzata opportunamente in funzione della tipologia di sacchetto da realizzare. L'attrezzaggio comporta un certo tempo di riconfigurazione (set-up) la cui entità dipende dalle dissimilarità tra le tipologie di due lotti produttivi consecutivi. Il problema consiste nel determinare l'ordinamento degli ordini di produzione in modo da terminare tutte le lavorazioni prima possibile.

# Introduzione alla teoria dello scheduling

## 1. Generalità

(continua)

### Esempi:

- Uno dei compiti di un sistema operativo è quello di disciplinare l'accesso alla CPU dei diversi programmi di calcolo. Ciascun programma può avere una certa priorità. L'obiettivo tipico del sistema operativo è allora quello di gestire l'insieme dei programmi in modo tale da minimizzare il tempo complessivo di attesa dei programmi, tenendo conto della loro importanza relativa. In questa particolare applicazione, il sistema operativo potrà eventualmente decidere di interrompere certi programmi per consentire il completamento di altri. Questa modalità operativa prende il nome di *preemption*.
- In un'officina di carrozzeria, vi sono quattro stazioni, dedicate rispettivamente a messa in forma, ribattitura, verniciatura, essiccazione a forno. In ciascuna stazione è attivo un operaio, che può lavorare su una sola autovettura alla volta. In una data giornata di lavoro, devono essere riparate un certo numero di autovetture sinistrate, ciascuna delle quali richiede il servizio da parte di alcune stazioni, in un dato ordine (ad esempio non si può riverniciare la carrozzeria prima di avere aggiustato le parti danneggiate). Il problema consiste nel gestire le varie operazioni in modo da terminare tutte le lavorazioni nel minor tempo possibile.

# Introduzione alla teoria dello scheduling

## 2. Il contesto di riferimento

### a) Gestione della produzione

Lo sviluppo di modelli di sequenziamento e scheduling è fortemente dovuto alle problematiche della pianificazione e schedulazione della produzione

In termini generali, la produzione può essere definita come il processo di conversione delle materie prime in prodotti finiti.

I sistemi di produzione sono tipicamente composti da una vasta gamma di componenti che devono essere gestiti efficacemente ed efficientemente per produrre i prodotti finiti nelle giuste quantità, nei dovuti tempi e costi.

La gestione della produzione coinvolge un vasto numero di aspetti decisionali delle organizzazioni produttive.

# Introduzione alla teoria dello scheduling

## 2. Il contesto di riferimento (continua)

Le principali decisioni seguendo la tassonomia di Anthony circa i processi decisionali aziendali (pianificazione strategica, pianificazione tattica e controllo delle attività operative) sono:

### Pianificazione strategica (politiche manageriali)

- Progettazione degli impianti di produzione
- Localizzazione di nuovi impianti
- Acquisizione di nuovi apparati
- Selezione di nuove linee di produzione
- Progettazione di sistemi logistici
- ...

Grandi investimenti, effetti a lunga scadenza, dati aggregati tipicamente esterni, alti fattori di rischio

### Pianificazione tattica (utilizzo delle risorse)

- Allocazione della capacità
- Pianificazione della produzione
- Definizione di canali distributivi
- Selezione di alternative di trasporto
- ...

Effetti a media scadenza, dati interni/esterni aggregati.

# Introduzione alla teoria dello scheduling

## 2. Il contesto di riferimento

(continua)

### Controllo delle attività operative (scheduling delle attività)

- Scheduling della produzione
- Controllo delle attività d'inventario
- Scheduling e routing di veicoli per la distribuzione
- Selezione di alternative di trasporto
- ...

Effetti a breve scadenza, dati interni disaggregati.

Ricapitolando:

FATTORI	PIANIFICAZIONE STRATEGICA	PIANIFICAZIONE TATTICA	CONTROLLO OPERATIVO
SCOPI	GESTIONE DEL CAMBIAMENTO	UTILIZZO RISORSE	ESECUZ., VAL., CONTROLLO
ORIZZ. PIAN.	LUNGO	MEDIO	BREVE
LIV. GEST.	ALTO	MEDIO	BASSO
FREQUENZA	BASSA	MEDIA	ALTA
DATI INPUT	ESTERNI	EST./INT.	INTERNI
AGGREGAZ.	ALTA	MODERATA	BASSA
INCERTEZZA	ALTA	MEDIA	BASSA
RISCHIO	ALTO	MEDIO	BASSO



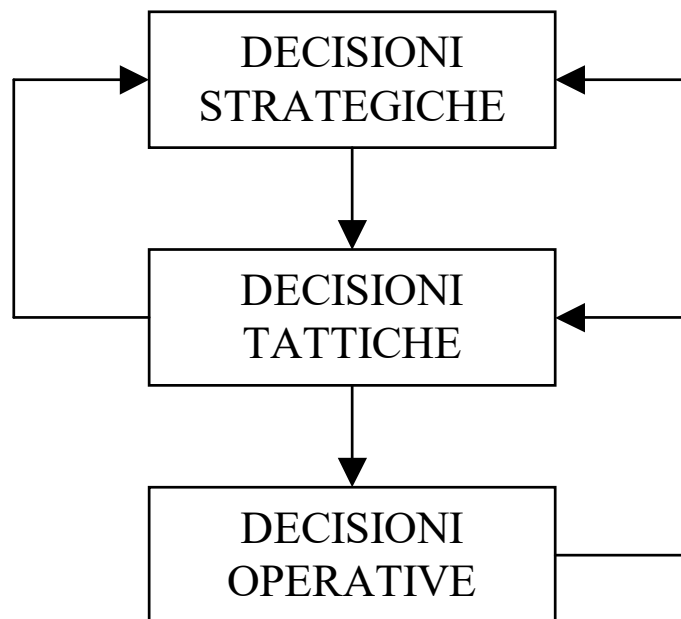
# Introduzione alla teoria dello scheduling

## 2. Il contesto di riferimento

(continua)

L'interdipendenza tra queste classi di decisioni è elevata, e spesso si ricorre ad approcci gerarchici in cui si cerca di ottimizzare i sottoproblemi

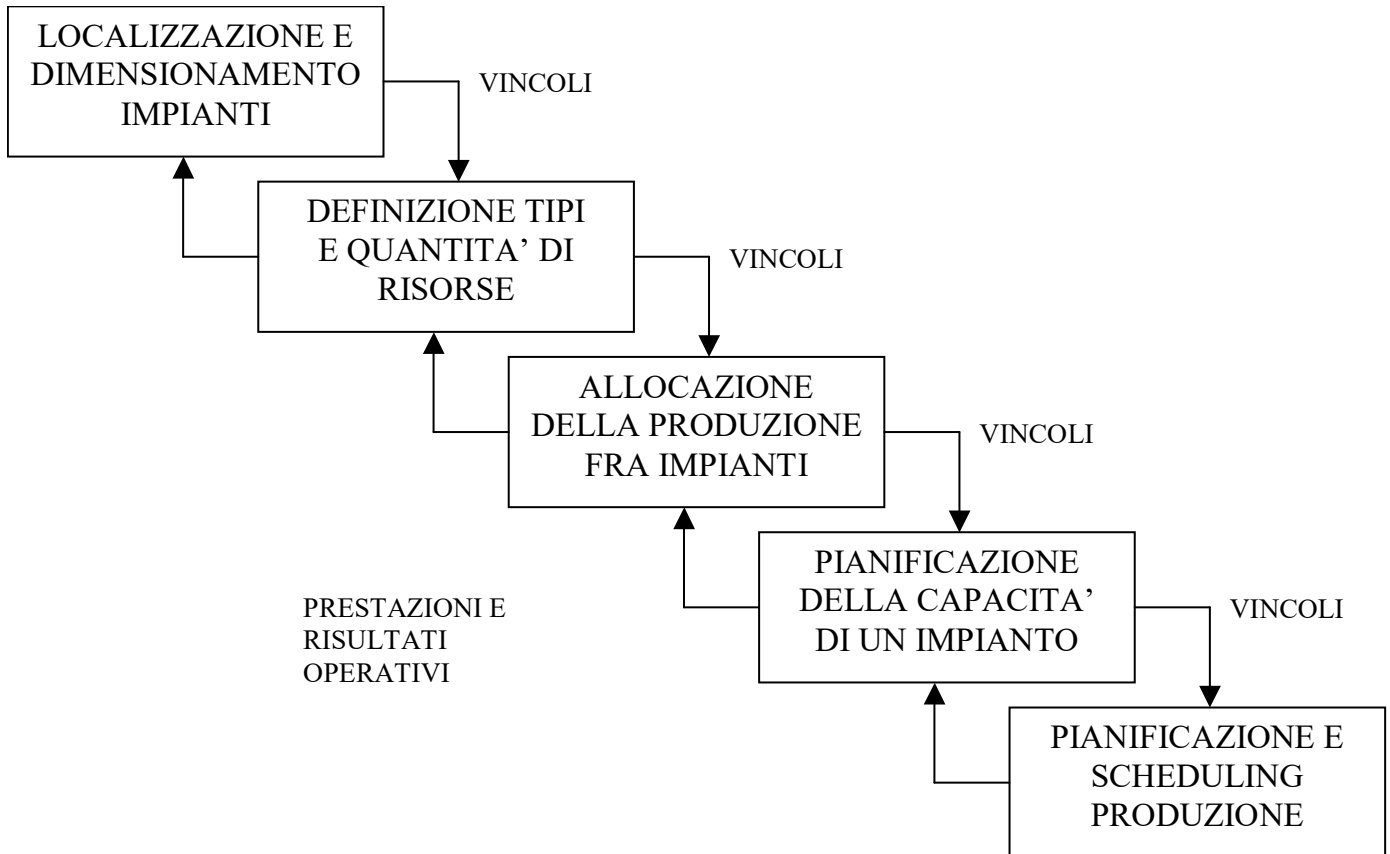
Secondo un approccio gerarchico, le decisioni sono decomposte in sottoproblemi, dove le decisioni ad alto livello (strategico) sono legate a quelle di livello inferiore e viceversa.



# Introduzione alla teoria dello scheduling

## 2. Il contesto di riferimento

(continua)



Le decisioni strategiche e tattiche impongono vincoli sulle decisioni operative.

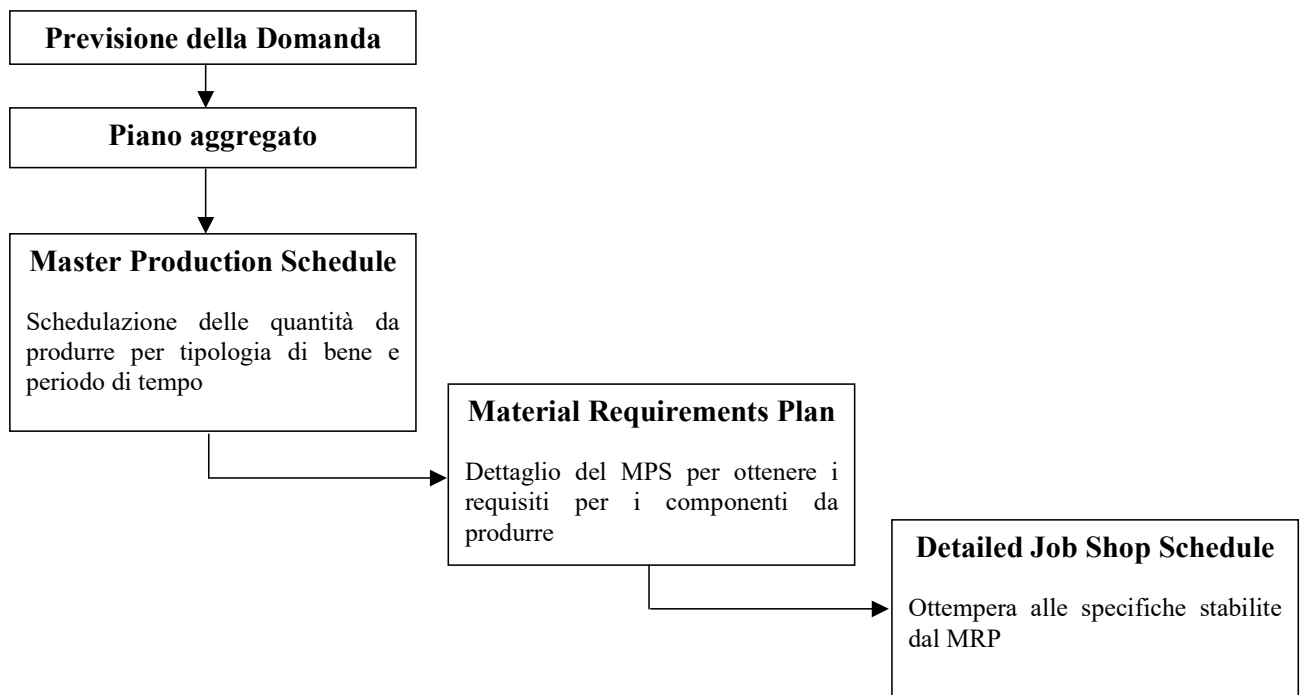
Le decisioni (dettagliate) operative forniscono informazioni per valutare la qualità delle decisioni aggregate di livello strategico e tattico.

# Introduzione alla teoria dello scheduling

## 2. Il contesto di riferimento

(continua)

### Gerarchia dei processi decisionali nella pianificazione e scheduling della produzione



- **MRP (Material Requirements Plan)**: fornisce le date di rilascio degli ordini e le date di consegna per ogni prodotto
- **Conseguenza**: diversi job possono competere per le stesse risorse (macchine, personale, ecc.) durante lo stesso periodo di tempo
- **Shop Scheduling**: fornire una schedulazione dettagliata delle operazioni in modo da rispettare i requisiti definiti dal MRP

# Introduzione alla teoria dello scheduling

## 2. Il contesto di riferimento

(continua)

### Pianificazione della produzione

E' il processo di:

- Determinazione di un piano per la produzione nel prossimo orizzonte temporale
- Determinazione dei livelli attesi d'inventario
- Definizione delle risorse necessarie ad implementare i piani produttivi
- ...

Viene condotta impiegando una vista aggregata dell'impianto di produzione, della domanda di prodotti

È influenzata dalle decisioni strategiche che vincolano la capacità produttiva

Unità temporale: settimane, mesi

Orizzonte temporale: frazione o unità di anni

Dati: aggregati per descrivere grandi porzioni dell'ambiente produttivo

# Introduzione alla teoria dello scheduling

## 2. Il contesto di riferimento

(continua)

### Scheduling della produzione

E' il processo di:

- Determinazione di un piano dettagliato di produzione per un breve orizzonte temporale

Viene condotta impiegando dati dettagliati sul processo produttivo

È influenzata dalle decisioni tattiche che fissano ad esempio le *release date* e *due date* per ciascun prodotto e altri vincoli temporali

Unità temporale: minuti

Orizzonte temporale: ore

Dati: dettagliati per descrivere piccole porzioni dell'ambiente produttivo

# Introduzione alla teoria dello scheduling

## 2. Il contesto di riferimento

(continua)

### Dispatching

E' il processo di:

- implementazione e controllo in tempo reale dei piani produttivi dettagliati

Cerca di far fronte a cambiamenti dell'ultimo momento dovuti a inconvenienti imprevisti

Gli argomenti del corso sono in gran parte dedicati all'analisi dei modelli a algoritmi applicabili allo scheduling della produzione e al dispatching.

# Introduzione alla teoria dello scheduling

## 3. Il processo Job-Shop

Nella letteratura con riferimento ai problemi di scheduling si fa riferimento al processo *job-shop* e si usa la terminologia dell'*ambiente manifatturiero*:

*job*, *macchina*, *operazione*, *tempo di processamento*, *ecc.*

L'*unità fondamentale* del processo di job-shop è l'*operazione* intesa come *task elementare* da eseguire. Ogni *operazione* è contraddistinta da tre *attributi*:

1. Un simbolo che ne identifica l'*appartenenza ad un job*;
2. Un simbolo che ne identifica l'*assegnamento ad una particolare macchina*;
3. Un numero reale che ne identifica il *tempo di esecuzione*.

L'insieme di tutte le operazioni di un processo job-shop può essere partizionato sulla base del primo simbolo in *sottoinsiemi disgiunti*, *esaustivi* e *mutuamente escludenti* detti *job*.

# Introduzione alla teoria dello scheduling

## 3. Il processo Job-Shop

(continua)

In relazione ad alcuni vincoli tecnologici in ogni *job* vi può essere un *ordinamento parziale* tra le sue operazioni rappresentabile da una *relazione di precedenza* che gode della proprietà transitiva. Così se  $x$  ed  $y$  sono operazioni dello stesso job ed  $x$  deve precedere  $y$  si scrive  $x < y$  nel caso di precedenza indiretta e  $x \prec y$  nel caso di precedenza diretta. Tipicamente la precedenza  $x < y$  implica che l'esecuzione di  $y$  non può iniziare prima del completamento di  $x$ . Se  $x \not< y$  e  $y \not< x$  le operazioni sono dette *indipendenti*.

Una *macchina* in questo processo è un *dispositivo in grado di eseguire un'operazione*, ovvero in senso astratto una *scala temporale* con certi intervalli di disponibilità.

Poiché le *operazioni* possono essere *partizionate sulla base del secondo indice in sottoinsiemi* associati alle macchine, allora un *job-shop* è *l'insieme di tutte le macchine ognuna delle quali identificate con un particolare insieme di operazioni*.

Un processo *job-shop* consiste delle *macchine*, dei *job*, e di una *regola* che limita il modo in cui le operazioni possono essere assegnate a specifici punti sulla scala temporale delle varie macchine.



# Introduzione alla teoria dello scheduling

## 3. Il processo Job-Shop

(continua)

Lo *scheduling* di un *processo job-shop* consiste nello *assegnare* un *insieme di intervalli*  $[b_x^1, c_x^1[$ ,  $[b_x^2, c_x^2[$ , ...,  $[b_x^{q_x}, c_x^{q_x}[$  *sulla scala dei tempi della macchina specificata* a ciascuna *operazione*  $x$ , tali che:

1.  $(c_x^1 - b_x^1) + (c_x^2 - b_x^2) + \dots + (c_x^{q_x} - b_x^{q_x}) \geq$  *tempo di processamento operazione*;
2.
  - a)  $c_x^{q_x} \leq b_y^1$  per ogni operazione  $y$  dello stesso job tale che  $x < y$ ;
  - b) Per ogni coppia  $x, y$  di operazioni *indipendenti* di uno stesso job gli intervalli assegnati a  $x$  sono disgiunti da quelli assegnati a  $y$ .
3. Ogni intervallo  $[b_x^k, c_x^k[$  *ricade interamente in uno degli intervalli disponibili* per l'assegnamento ad una determinata macchina.

# Introduzione alla teoria dello scheduling

## 3. Il processo Job-Shop

(continua)

Facendo riferimento al caso in cui *non* è consentita la *preempzione* delle operazioni per cui un singolo intervallo  $[b_x, c_x[$  pari al suo tempo di processamento di è assegnato a ciascuna operazione  $x$ ,

alternativamente lo *scheduling* di un processo di *job-shop* può essere visto come la *costruzione* di un *ordinamento delle operazioni associate a ciascuna macchina*, con le seguenti *restrizioni*:

1. Ogni *macchina*, essendo disponibile con continuità può essere vista come un *singolo intervallo temporale illimitato superiormente*.
2. Ogni *operazione* può essere eseguita soltanto da una *macchina*.
3. *Nello shop* vi è solo una *macchina* di ogni tipo per cui i numeri di identificazione delle macchine sono unici ed identificano per ogni operazione nel job una sola macchina nello shop.
4. Le *operazioni*  $x$  e  $y$  di un job non possono essere *sovrapposte*, cioè gli intervalli  $(b_x, c_x)$   $(b_y, c_y)$  devono essere disgiunti, e  $b_y \geq c_x$  ( $b_x \geq c_y$ ) se  $x < y$  ( $y < x$ ).
5. Una *macchina* può eseguire un'operazione alla volta per cui se  $x$  e  $z$  devono essere eseguite dalla stessa macchina  $b_x \geq c_z$  ovvero  $c_x \leq b_z$ .

# Introduzione alla teoria dello scheduling

## 3. Il processo Job-Shop

(continua)

Nella descrizione del processo *job-shop* ora fatta è da sottolineare che esiste *una sola risorsa limitante*. Ogni operazione infatti viene eseguita da *una sola macchina*. Per questo motivo un tale *processo job-shop* è detto *semplice*.

Alternativamente si possono avere *job-shop* con *più risorse limitanti*. Infatti può accadere che l'esecuzione di *un'operazione richieda più macchine ovvero una macchina e altre risorse* (persone, equipaggiamento, ecc.) *contemporaneamente*.

In tal caso *l'assegnazione di un'operazione ad un punto della scala dei tempi non può essere fatta finché le risorse necessarie non sono contemporaneamente disponibili*.

# Introduzione alla teoria dello scheduling

## 4. Definizioni generali per i problemi di scheduling

Formalmente un problema generale di scheduling è caratterizzabile dai seguenti tre insiemi:

- $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$ , insieme di  $n$  task (*operazioni*)
- $\mathcal{P} = \{P_1, P_2, \dots, P_m\}$ , insieme di  $m$  processori (*macchine*)
- $\mathcal{R} = \{R_1, R_2, \dots, R_s\}$ , insieme di  $s$  tipi di *risorse aggiuntive*

In questo contesto la parola "*scheduling*" significa *assegnare macchine e (eventualmente) risorse aggiuntive ai task nel tempo* in modo da eseguirli tutti rispettando i *vincoli* che nella classica teoria dello scheduling sono tipicamente i seguenti:

1. *Ogni task viene eseguito (tipicamente) da una macchina (che può essere specifica) ed da altre risorse aggiuntive (se richieste) ad un certo istante;*
2. *Ogni macchina può eseguire al più un task alla volta.*

In generale si assume che i task siano partizionabili in sottoinsiemi (in generale catene) di operazioni non eseguibili simultaneamente (a causa di vincoli tecnologici), ciascuno dei quali è chiamato *job*.

Pertanto, un job  $j = \{T_{1j}, T_{2j}, \dots, T_{nj}\}$  e qualunque coppia di task di  $j$  richiedono (tipicamente) macchine distinte per la loro esecuzione.

# Introduzione alla teoria dello scheduling

## 4. Definizioni generali per i prob. scheduling (continua)

- Modelli di processamento

Con riferimento alle modalità di esecuzione dei task le *macchine* si distinguono in *due tipologie*:

a) *Parallele*: se svolgono la stessa funzione.

Le macchine parallele si distinguono inoltre sulla base della loro velocità di esecuzione in:

a1) *Identiche*: le macchine hanno tutte la *stessa velocità  $v$  di processamento*;

Il tempo di processamento di un task  $T_j$  dipende solo dal task e non dalla macchina  $P_i$  che lo esegue:  $p_{ij} = p_j = l_j/v$  ( $l_j$  lunghezza di  $T_j$ )

a2) *Uniformi*: le macchine hanno *diverse velocità  $v_i$  di processamento, ma invarianti rispetto ai task*;

Il tempo di processamento di un task  $T_j$  è inversamente proporzionale alla velocità  $v_i$  della macchina  $P_i$  che lo esegue:  $p_{ij} = l_j/v_i$

a3) *Non-correlate*: le *velocità  $v_{ij}$  delle singole macchine sono diverse e dipendenti dai task*.

Il tempo di processamento di un task  $T_j$  dipende in modo del tutto generale dal task  $a$  dalla macchina che lo esegue:  $p_{ij} = l_j/v_{ij}$

# Introduzione alla teoria dello scheduling

## 4. Definizioni generali per i prob. scheduling (continua)

b) **Dedicate**: se sono specializzate per l'esecuzione di particolari task.

In generale -modello **general job shop**- il numero  $n_j$  dei task di ogni job è arbitrario e non vi è un ordinamento prefissato per il loro processamento. in questo contesto si distinguono **tre sotto-modelli**:

b1) **Open-Shop**: il numero dei task è lo stesso per ogni job ossia  $n_j = K, \forall j$ , e non sono date relazioni di precedenza fra i task di un job.

b2) **Flow-Shop**: in aggiunta ad  $n_j = K, \forall j$ , il processamento di  $T_{i-1j}$  deve precedere quello di  $T_{ij}, \forall i, \forall j$ .

b3) **Job-Shop**: il numero  $n_j$  dei task di ogni job è arbitrario e vi è un ordinamento totale sui task di ogni job.

# Introduzione alla teoria dello scheduling

## 4. Definizioni generali per i prob. scheduling (continua)

Nel processo di scheduling è importante distinguere tra **parametri che definiscono il problema** noti a priori (nel caso deterministico) e **variabili che descrivono la soluzione** prodotta dallo scheduling:

a) **Parametri**: ad ogni job  $j$  sono associati i seguenti dati.

a1)  $r_j =$  **ready time** (o **release date**): indica il **tempo di arrivo nello shop** ossia il l'istante minimo d'inizio del processamento della prima operazione del job stesso;

a2)  $d_j =$  **due date**: indica il **tempo limite** entro cui ci si aspetta **termini il processamento del job**. Normalmente a seconda delle **due date** sono definite delle funzioni di penalità;

a3)  $p_j = \sum_{i=1}^{n_j} p_{ij}$  **processing time**: indica il **tempo di esecuzione di tutte le operazioni del job**, una volta dato il vettore dei tempi di processamento  $[p_{1j}, p_{2j}, \dots, p_{n_j}]^T$  associato allo insieme delle macchine  $\mathcal{P} = \{P_1, P_2, \dots, P_{n_j}\}$ .

a4)  $a_j = (d_j - r_j)$  **allowance**: indica il **tempo totale a disposizione del job nello shop**.

# Introduzione alla teoria dello scheduling

## 4. Definizioni generali per i prob. scheduling (continua)

b) **Variabili**: la soluzione di un problema di scheduling comporta la *determinazione del tempo in cui ogni job deve essere processato* e più precisamente *quando ogni operazione deve essere eseguita*. Ciò significa *determinare* per ogni operazione il valore della *seguinte variabile*

$W_{ij}$ : *waiting time* della  $i$ -ma operazione  $T_{ij}$  del job  $j$ , cioè il tempo di attesa nello shop dopo la  $(i-1)$ -ma operazione prima che abbia inizio la  $i$ -ma operazione.

Il *tempo totale di attesa del job* è:

$$W_j = \sum_{i=1}^{n_j} W_{ij}$$

La *schedula* di un particolare problema è pertanto completamente *specificata* conoscendo l'insieme  $\{W_{ij}\}$ .

Ogni altra variabile introdotta per valutare la bontà di una schedula è funzione di  $W_{ij}$ :



# Introduzione alla teoria dello scheduling

## 4. Definizioni generali per i prob. scheduling (continua)

1.  $C_j$ : **tempo di completamento del job  $j$** ; indica il **l'istante di tempo in cui l'ultima operazione  $T_{nj}$  del job  $j$  è completata.**

$$C_j = r_j + W_{1j} + p_{1j} + \dots + W_{nj} + p_{nj} = r_j + p_j + W_j$$

2.  $F_j$ : **flow-time del job  $j$** ; indica il **tempo totale trascorso nello shop dal job  $j$**

$$F_j = W_{1j} + p_{1j} + \dots + W_{nj} + p_{nj} = p_j + W_j = C_j - r_j$$

3.  $L_j$ : **lateness del job  $j$**

$$L_j = C_j - d_j = F_j - a_j$$

4.  $T_j$ : **tardiness del job  $j$**

$$T_j = \max\{0, L_j\}$$

5.  $E_j$ : **earliness del job  $j$**

$$E_j = \max\{0, -L_j\}$$

6.  $U_j$ : **job  $j$  in ritardo**

$$U_j = 1, \text{ se } C_j > d_j, \text{ altrimenti } U_j = 0$$

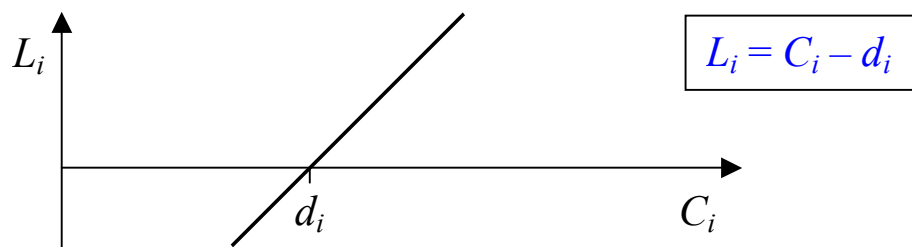
Le ultime quattro variabili rappresentano differenti **modalità di confronto tra il tempo di completamento e la due-date.**

# Introduzione alla teoria dello scheduling

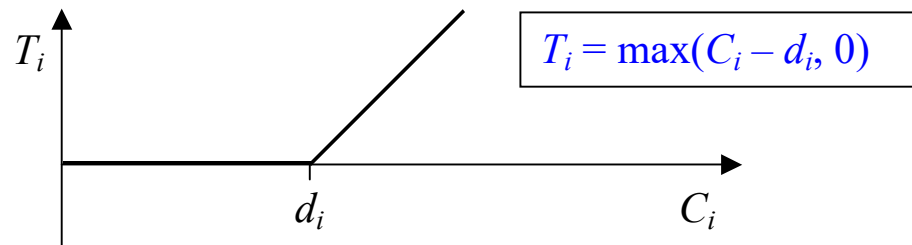
## 4. Definizioni generali per i prob. scheduling (continua)

In particolare:

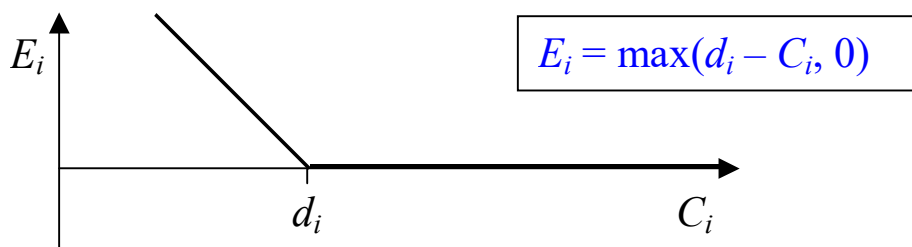
$L_j$  = differenza algebrica; indica lo **scostamento tra tempo di completamento reale e desiderato**;



$T_j$  = **ritardo** rispetto alla due-date;



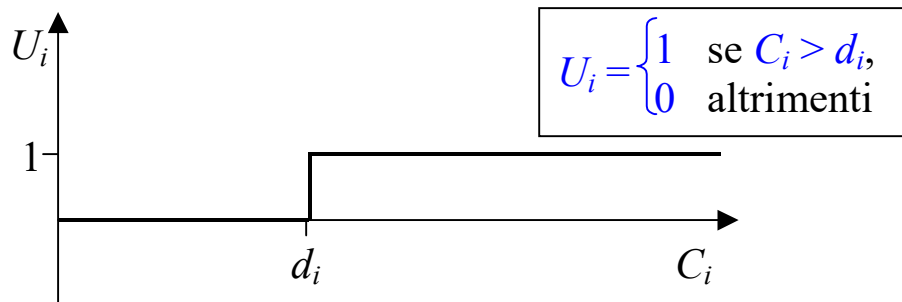
$E_j$  = **anticipo** rispetto alla due-date;



# Introduzione alla teoria dello scheduling

## 4. Definizioni generali per i prob. scheduling (continua)

$U_j =$  job in ritardo (o meno).



Queste variabili di output possono essere usate per costruire degli *indicatori di prestazioni* delle schedale:

*Makespan (massimo tempo di completamento), tempo di completamento totale, flow time medio, massima lateness, massima tardiness, numero di job in ritardo, ecc.*

c) *Caratteristiche* generali delle schedale:

Una *schedula* e' detta *preemptiva* se il *processamento di ogni task può essere interrotto in qualsiasi istante e ripreso più tardi.*

*Se l'interruzione del processamento dei task non è consentita la *schedula* si dice *non-preemptiva.**

# Introduzione alla teoria dello scheduling

## 5. Criteri di valutazione delle schedule

- Tipicamente gli indicatori di prestazione usati sono misure espresse come

$$M = f(C_1, C_2, \dots, C_n)$$

- L'obiettivo è determinare schedule che ottimizzino tali indicatori, ad esempio minimizzare una data funzione di misura  $f(C_1, C_2, \dots, C_n)$
- Le *misure di prestazione più frequentemente* adottate nello scheduling sono quelle esprimibili attraverso *funzioni regolari*.

*Def.:*  $f$  è *regolare* se è *non decrescente*, cioè se

$$C'_i > C_i \text{ e } C'_j \geq C_j, \text{ per } 1 \leq j \neq i \leq n \rightarrow M' = f(C'_1, C'_2, \dots, C'_n) \geq M$$

- Il *valore medio o massimo della "earliness"* *non* è una misura *regolare* (se  $C_j$  aumenta  $E_j$  diminuisce), mentre *lo sono* il *valor medio* ed il *valor massimo* del: *tempo di completamento, flow time, lateness, e tardiness*.

# Introduzione alla teoria dello scheduling

## 5. Criteri di valutazione delle schedule (continua)

### Esempi di funzioni di misura da minimizzare

- **Makespan**  $C_{\max} = \max_j \{C_j\}$

Misura il tempo di completamento del job che termina per ultimo.

Minimizzare il makespan comporta un elevato utilizzo delle macchine e bilancia il loro impiego.

- **Massima lateness**  $L_{\max} = \max_j \{L_j\} = \max_j \{C_j - d_j\}$

Misura la massima violazione delle due-date.

- **Massima tardiness**  $T_{\max} = \max_j \{T_j\} = \max_j \{\max[0, C_j - d_j]\}$

Misura il ritardo del job completato con maggior ritardo rispetto alla sua due-date.

Minimizzare tale funzione significa minimizzare il ritardo accusato dal job più penalizzato dal punto di vista dei ritardi.

# Introduzione alla teoria dello scheduling

## 5. Criteri di valutazione delle schedule (continua)

- **Tempo di completamento pesato totale**  $\sum_j w_j C_j$

Fornisce un'indicazione del tempo medio pesato di permanenza dei job nel sistema di processamento; misura il livello di servizio offerto dal sistema. Minimizzare tale funzione significa minimizzare il costo medio pesato dei job se questo è legato alla loro attesa.

- **Ritardo totale pesato**  $\sum_j w_j T_j$

Misura il ritardo il ritardo complessivo pesato dei job. Minimizzare tale funzione comporta minimizzare il costo totale dovuto ai ritardi.

- **Numero pesato di job in ritardo**  $\sum_j w_j U_j$

Misura il peso (importanza) totale dei job terminati in ritardo. Con pesi unitari, fornisce un'indicazione della percentuale di job eseguiti in tempo.

Minimizzare tale funzione comporta la massimizzazione del numero (pesato) dei job completati in tempo.

# Introduzione alla teoria dello scheduling

## 5. Criteri di valutazione delle schedule (continua)

- Esaminiamo la **correlazione** tra alcune misure regolari. Poiché per ogni job vale:

$$L_j = F_j - a_j = C_j - r_j - a_j = C_j - d_j$$

Per  $n$  job si ha

$$\sum L_j = \sum F_j - \sum a_j = \sum C_j - \sum r_j - \sum a_j = \sum C_j - \sum d_j$$

Ossia per i valori medi

$$\bar{L} = \bar{F} - \bar{a} = \bar{C} - \bar{r} - \bar{a} = \bar{C} - \bar{d}$$

Poiché  $\bar{a}$ ,  $\bar{r}$ , e  $\bar{d}$  sono costanti per un dato problema ne consegue che **schedule ottime rispetto a  $\bar{L}$  lo sono anche rispetto a  $\bar{C}$  e  $\bar{F}$ .**

- Poiché

$$C_j = r_j + W_j + p_j$$

ne consegue che

$$\bar{C} = \bar{r} + \bar{W} + \bar{p}$$

ossia, essendo  $\bar{r}$  e  $\bar{p}$  costanti, **schedule ottime rispetto a  $\bar{C}$  lo sono anche rispetto a  $\bar{W}$ .**

# Introduzione alla teoria dello scheduling

## 5. Criteri di valutazione delle schedule (continua)

- Inoltre poiché

$$T_j - E_j = L_j \Rightarrow \bar{T} - \bar{E} = \bar{L}$$

questo *non implica* che *schedule* con  $\bar{L}$  *minima* abbiano anche  $\bar{T}$  *minima*.

*Solo se* i job hanno due-date tali che tutti gli  $L_j$  sono *positivi*,  $\bar{L}$  e  $\bar{T}$  sono *misure equivalenti*.

*Osservazione.* Quanto detto per i valori medi *non può essere esteso ai valori massimi*.

Se due schedule sono equivalenti rispetto a  $\bar{C}$ ,  $\bar{F}$  e  $\bar{L}$  nessuna conclusione si può ricavare per  $C_{\max}$ ,  $F_{\max}$  ed  $L_{\max}$ .



# Introduzione alla teoria dello scheduling

## 6. Classificazione dei problemi di scheduling

- Un problema di scheduling è descritto da quattro tipi di informazioni:
  - 1) i job e le operazioni da processare;
  - 2) il numero ed il tipo di macchine presenti nello shop nonché le regole che limitano gli assegnamenti dei job alle macchine;
  - 3) le caratteristiche dei job e delle risorse aggiuntive;
  - 4) i criteri per valutare la bontà di una schedula.
- Per quanto riguarda il punto 1) occorre distinguere i job in base alla natura degli arrivi nello shop.

Nel caso *statico* un certo numero di job presenti nello shop ad un certo istante viene processato senza che altri job arrivino durante l'esecuzione del processo.

Nel caso *dinamico* i job arrivano continuamente ed i tempi di arrivo possono essere previsti solo su base statistica.
- Il punto 2) definisce l'*ambiente di processamento*. Occorre distinguere due casi estremi. Il primo è il *flow-shop* in cui tutti i job seguono lo stesso cammino da una macchina all'altra. L'altro è il *job-shop* in cui non vi è un comune movimento da una macchina all'altra.

# Introduzione alla teoria dello scheduling

## 6. Classificazione dei problemi di scheduling (continua)

- Il punto 3) definisce le **caratteristiche dei job**. Si possono distinguere i casi di **risorse addizionali o meno**, **preempzione o non preempzione**, **ready time uguali o meno** e così via.
- Il punto 4) definisce l'**indice di prestazione** scelto rispetto a cui ottimizzare la schedula.
- Pertanto i **problemi di scheduling** possono essere **classificati** mediante una **tassonomia** basata su **quattro parametri** usando la **notazione**  $*|\alpha|\beta|\gamma$  corrispondente ai quattro tipi di informazioni descritte prima.
- Nel **caso dinamico** il simbolo  $*$  identifica la distribuzione di probabilità dei tempi di arrivo. Nel **caso statico (deterministico)**  $*$  specifica solo il numero dei job da processare e quindi non serve a caratterizzare il problema; in questo caso si usano soltanto i **tre campi**  $\alpha|\beta|\gamma$ .

# Introduzione alla teoria dello scheduling

## 6. Classificazione dei problemi di scheduling (continua)

- Esempi:

(i)  $P || C_{\max}$  questo problema indica lo *scheduling non-preemptivo di job indipendenti* ed arbitrari, *disponibili al tempo 0*, su *macchine parallele identiche* con l'obiettivo di *minimizzare la lunghezza della schedula (makespan)*.

(ii)  $Q |_{\text{prec}} | L_{\max}$  questo problema indica lo *scheduling non-preemptivo di job dipendenti* ed arbitrari, *disponibili al tempo 0*, su *macchine parallele uniformi* con l'obiettivo di *minimizzare la lateness massima*.

(iii)  $O |_{\text{pmtn}, r_j} | \sum C_j$  *scheduling preemptivo di job indipendenti* ed arbitrari che *arrivano in un "open shop" a istanti di tempo diversi* con l'obiettivo di *minimizzare il flow time totale*.

(iv)  $J |_{r_j} | \sum T_j$  *scheduling non-preemptivo di job indipendenti* ed arbitrari che *arrivano in un "job shop" a istanti diversi* con l'obiettivo di *minimizzare la tardiness totale*.

# Introduzione alla teoria dello scheduling

## 7. Rappresentazione dei problemi di scheduling

- Dato un problema generale di scheduling i *job* sono detti "*indipendenti*" se tra di essi *non esiste alcuna relazione di precedenza*.
- Se viceversa i *job* sono (parzialmente) *ordinati da una relazione di precedenza* essi sono detti "*dipendenti*" e questa situazione può essere *rappresentata* attraverso un *grafo orientato*.

Si possono avere *due tipi di rappresentazioni*:

- (i) *grafi con job sui nodi*
- (ii) *grafi con job sugli archi*

Nel caso (i) i nodi corrispondono ai job da eseguire e gli archi rappresentano i vincoli di precedenza.

Nel caso (ii) gli archi rappresentano i job ed i nodi corrispondono ad eventi temporali (*inizio o termine di un'attività*).

- Le *schedule* soluzioni del problema possono essere *rappresentate* attraverso *diagrammi di Gantt*.

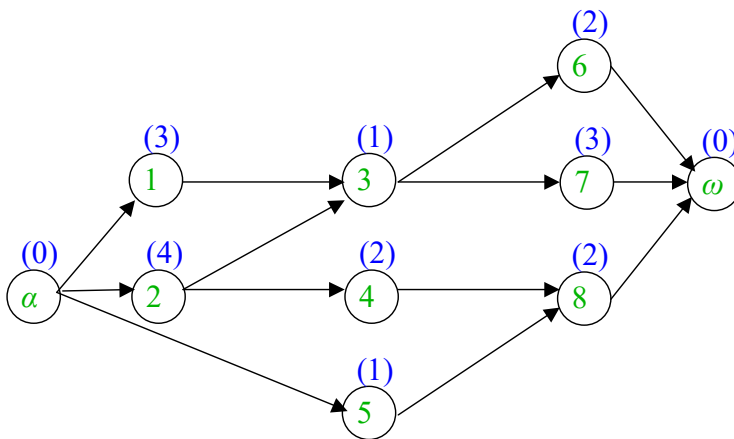
# Introduzione alla teoria dello scheduling

## 7. Rappresentazione dei problemi di scheduling (continua)

Data la seguente istanza con 8 job del problema  $P3 | \text{prec} | C_{\max}$

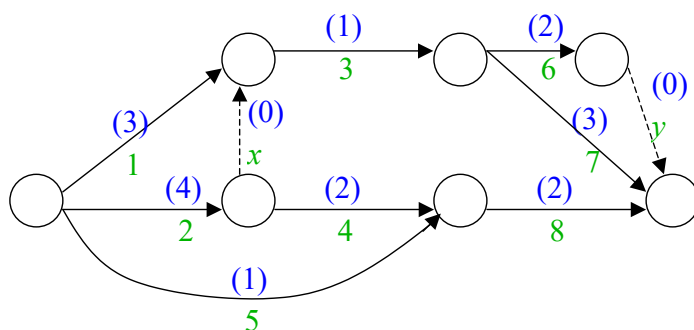
job	1	2	3	4	5	6	7	8
$p_i$	3	4	1	2	1	2	3	2

Con le relazioni di precedenza rappresentate da:



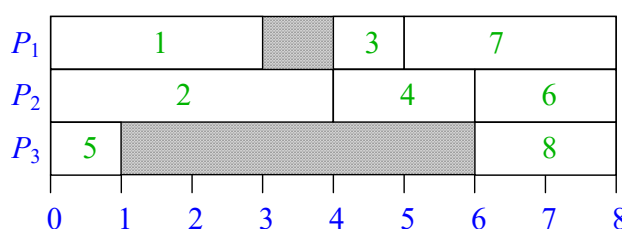
Grafo con attività sui nodi

ovvero da:



Grafo con attività sugli archi

una schedula ammissibile è



Schedula ammissibile

# Introduzione alla teoria dello scheduling

## 8. Classi di schedule

- Nella terminologia della teoria dello scheduling si usa spesso la distinzione tra: *sequenza*, *schedula* e *politica di scheduling*.
- Una *sequenza* indica una *permutazione dei job* corrispondente all'ordine con cui i job devono essere processati su una data macchina.
- Una *schedula* usualmente indica un'*allocazione nel tempo dei job da processare in un contesto di macchine più complesso*, in cui è consentita la preempzione di un job se un altro job arriva più tardi nel sistema.
- Il concetto di *politica di scheduling* è usato generalmente in un contesto stocastico in cui viene prescritta un'azione appropriata per ognuno degli stati in cui il sistema viene a trovarsi.
- Pertanto nei *modelli deterministici* usualmente sono rilevanti solo *sequenze* e *schedule*.
- Con riferimento alle schedule sono rilevanti le seguenti tre classi (introdotte da Baker, 1974):
  - (i) *schedule senza ritardo (non-delay)*
  - (ii) *schedule attive*
  - (iii) *schedule semi-attive*

# Introduzione alla teoria dello scheduling

## 8. Classi di schedule

(continua)

### (i) *Schedule senza ritardo*

- Una schedula ammissibile è detta **senza ritardo** (**non-delay**) se nessuna macchina è lasciata libera quando è disponibile un'operazione per essere processata. Pertanto richiedere che una schedula sia non-delay implica che **non** viene **consentita** una **inattività non necessaria**.
- In generale nella maggior parte dei modelli con misura di prestazione regolare le schedule ottime sono non-delay. Tuttavia in alcuni casi può essere più conveniente consentire periodi di inattività non necessaria (modelli non-preemptivi).

### (ii) *Schedule attive*

- Una schedula ammissibile è detta **attiva** se nessuna operazione può essere completata prima, anche alterando l'ordine di esecuzione delle operazioni sulle macchine, senza ritardare nessuna altra operazione. Una **schedula non-delay è certamente attiva** ma **non è vero il viceversa**. Pertanto la classe delle schedule attive è più ampia di quella delle schedule non-delay.

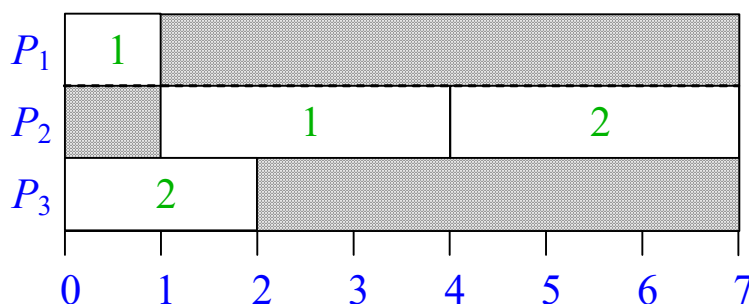
# Introduzione alla teoria dello scheduling

## 8. Classi di schedule

(continua)

**Esempio:** Si consideri un job-shop con tre macchine e due job. Il job 1 richiede un tempo 1 sulla macchina 1 ed un tempo 3 sulla macchina 2; il job 2 richiede un tempo 3 sulla macchina 2 e un tempo 2 sulla macchina 3; entrambi i job devono essere processati per ultimo sulla macchina 2.

Consideriamo la schedula in cui sulla macchina 2 il processamento del job 1 precede quello del job 2



Schedula  
senza-ritardo

Ovviamente è anche attiva perché scambiando l'ordine di esecuzione delle attività dei job 1 e 2 sulla macchina 2 consente di completare prima l'operazione del job 2 ma ritarda quella del job 1.

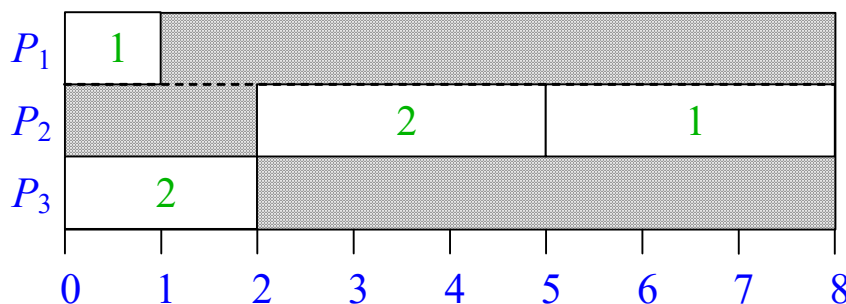


# Introduzione alla teoria dello scheduling

## 8. Classi di schedule

(continua)

Consideriamo la schedula in cui sulla **macchina 2** il processamento del **job 2** precede quello del **job 1**



Schedula attiva  
ma con ritardo

Scambiando la sequenza dei due job sulla macchina 2 viene ritardato il processamento del job 2.

Pertanto la **schedula** è **attiva** ma **non è non-delay**.  
Infatti la macchina 2 resta inattiva fino all'istante 2 anche se all'istante 1 è disponibile il job 1.

### (iii) **Schedule semi-attive**

- Una schedula ammissibile è detta **semi-attiva** se nessuna operazione può essere completata prima senza modificare la sequenza di processamento su una qualsiasi delle macchine dello shop.

# Introduzione alla teoria dello scheduling

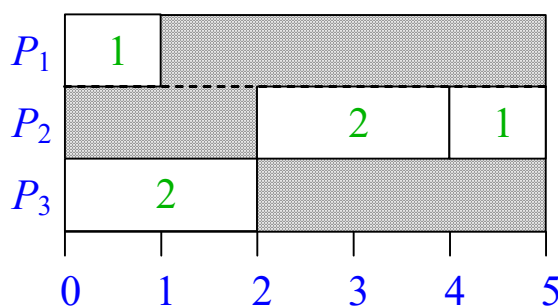
## 8. Classi di schedule

(continua)

Una **schedula attiva è semi-attiva** ma **non è vero il viceversa**. Quindi la classe delle schedule semi-attive è più ampia di quella delle schedule attive.

**Esempio:** job-shop con tre macchine e due job. L'ordine di processamento sulle macchine è lo stesso del precedente esempio. Il **job 1** richiede un **tempo 1** sia sulla **macchina 1** che sulla **2**; il **job 2** richiede un **tempo 2** sia sulla **macchina 2** che sulla **3**.

Consideriamo la schedula in cui il processamento sulla **macchina 2** del **job 2** avviene prima del **job 1**



**Schedula semi-attiva  
ma non attiva  
e quindi anche con  
ritardo**

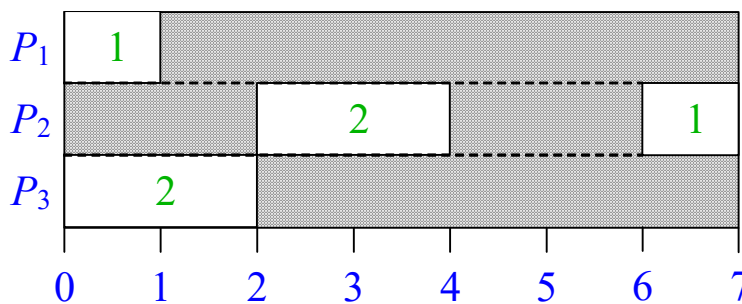
Il processamento del **job 1** può essere completato prima scambiando la sequenza dei due job sulla **macchina 2** senza ritardare il **job 2**.

# Introduzione alla teoria dello scheduling

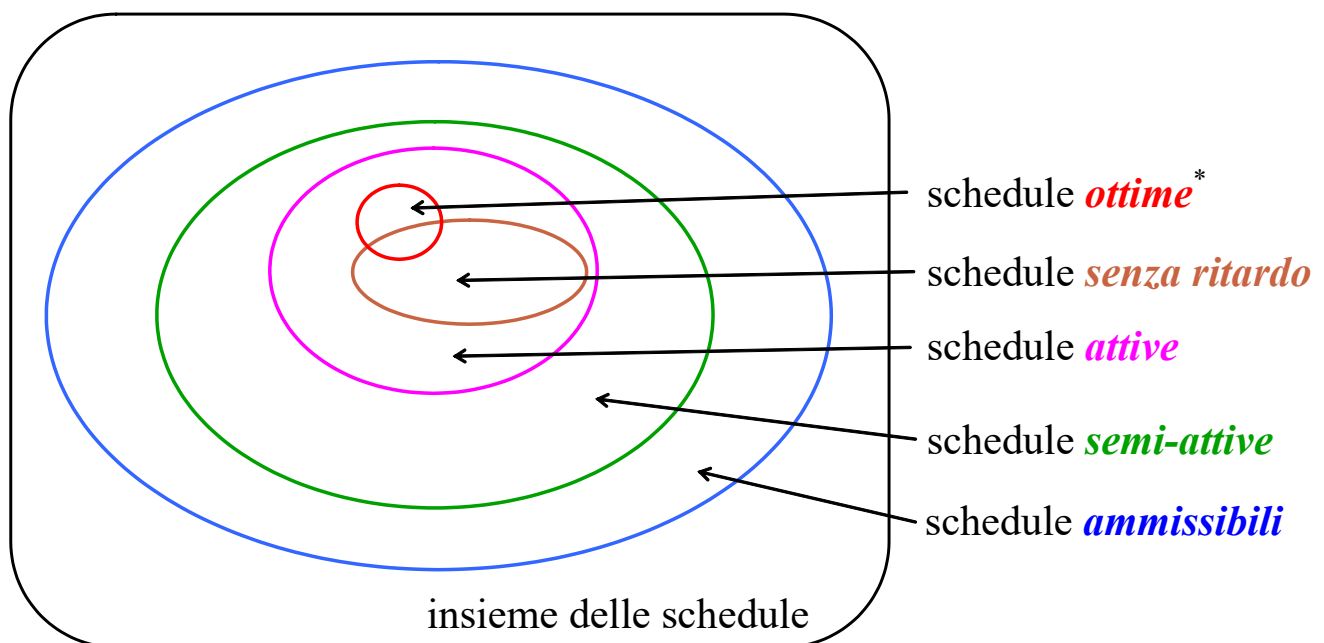
## 8. Classi di schedule

(continua)

Una **schedula non semi-attiva** si può costruire per esempio ritardando il job 1 sulla macchina 2 di una unità.



Schedula  
non semi-attiva



\* Nel caso di misura di prestazione regolare

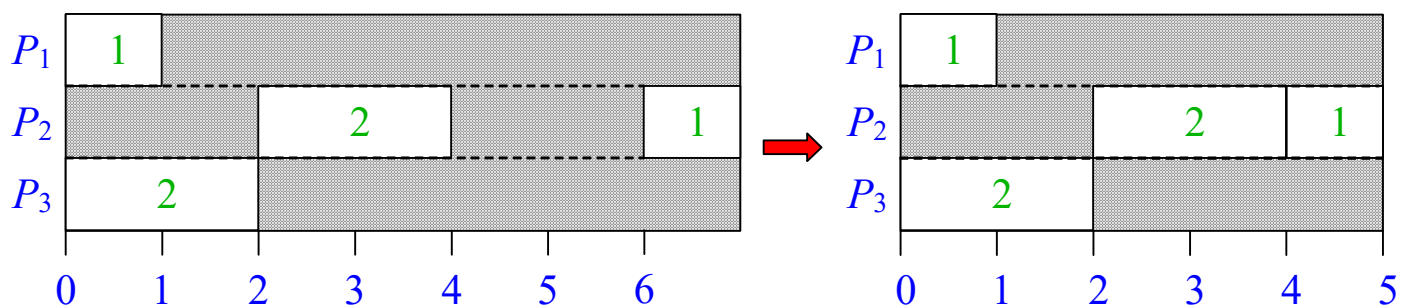
# Introduzione alla teoria dello scheduling

## 8. Classi di schedule

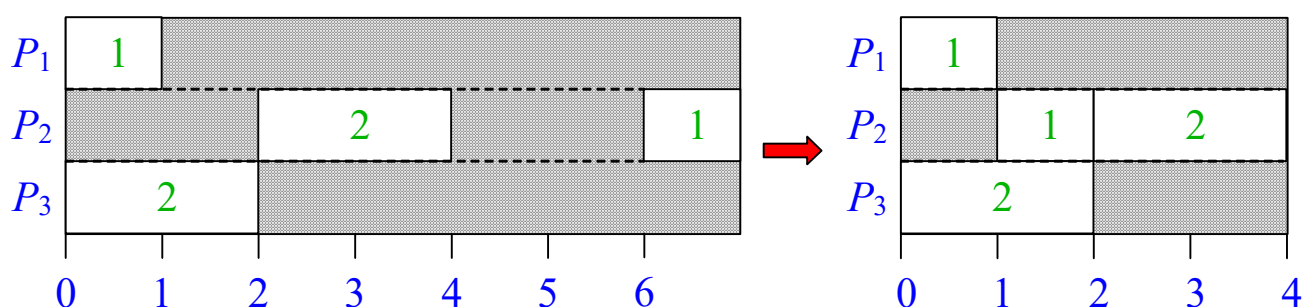
(continua)

- Il concetto di schedula attiva e semi-attiva è stato esteso al caso di problemi più generali (RCPSP) (Sprecher et al., 1995) introducendo il concetto di *local left-shift* e di *global left-shift*:

Data una schedula ammissibile  $S$ , un *local left-shift* di un'operazione  $T_{ij}$  consiste nell'anticipare di  $h \geq 1$  unità di tempo l'inizio di  $T_{ij}$ , attraverso  $h$  *left-shift unitari ammissibili*, ottenendo una schedula  $S'$  ammissibile.



Data una schedula ammissibile  $S$ , un *global left-shift* di un'operazione  $T_{ij}$  consiste nell'anticipare di  $h > 1$  unità di tempo il tempo d'inizio di  $T_{ij}$ , ottenendo una schedula  $S'$  ammissibile *non ottenibile attraverso un local left-shift*



# Introduzione alla teoria dello scheduling

## 8. Classi di schedule

(continua)

- Una schedula **semi-attiva** è una schedula ammissibile in cui **nessuna** operazione può essere anticipata con un **local left-shift**.
- Una schedula **attiva** è una schedula ammissibile (e semi-attiva) in cui **nessuna** operazione può essere anticipata con un **local left-shift** o con un **global left-shift**.
- Una schedula **non-delay** è una schedula ammissibile attiva anche per il caso preemptivo.

### Algoritmo per generare tutte le schedule attive

- Sia  $\Omega$  l'insieme delle operazioni schedulabili (perché tutti gli eventuali predecessori sono stati schedulati).
- Sia  $p_{ij}$  il tempo di processamento e  $r_{ij}$  l'earliest starting time dell'operazione  $T_{ij} \in \Omega$  del job  $j$  sulla macchina  $i$ .

**Step 1** Sia  $\Omega$  l'insieme delle prime operazioni schedulabili (prive di predecessori). Sia  $r_{ij} = r_j$ , per tutte le operazioni  $T_{ij} \in \Omega$ .

**Step 2** Calcola  $t(\Omega) = \min_{T_{ij} \in \Omega} \{r_{ij} + p_{ij}\}$  e sia  $i^*$  la macchina per la quale si ottiene il minimo.

**Step 3** Sia  $\Omega' \subseteq \Omega$  l'insieme di tutte le operazioni  $T_{i^*j}$  sulla macchina  $i^*$  tale che  $r_{i^*j} < t(\Omega)$ .

**Step 4** Per ogni operazione  $T_{i^*j} \in \Omega'$ , poni (alternativamente)  $T_{i^*j}$  come la successiva sulla macchina  $i^*$ , schedulandola al tempo  $r_{i^*j}$ . Per ognuna di queste scelte, rimuovi l'operazione  $T_{i^*j}$  da  $\Omega$  e aggiungi ad  $\Omega$  tutti i suoi immediati successori, calcolandone gli earliest start time e vai allo step 2.

Per le schedule **non-delay** considerare nello step 2  $t(\Omega) = \min_{T_{ij} \in \Omega} \{r_{ij}\}$  e nello step 3 la condizione  $r_{i^*j} = t(\Omega)$ . Per le **semi-attive** non si eseguono gli step 2 e 3 e si considera  $\Omega' = \Omega$  e  $i^*$  una qualsiasi delle macchine.