

Problemi di scheduling

Indice

Perché lo scheduling su macchine parallele?

Panoramica della famiglia

Scheduling su macchine parallele identiche

Contare le soluzioni

Un primo schedule a mano

Un lower bound combinatorio

Relazione con il problema del bin packing

Osservazioni

Errori tipici

Obiettivi

Al termine di questo capitolo saremo in grado di:

- ▶ classificare un problema di scheduling con la notazione a tre campi ed enunciare il problema $P \parallel C_{\max}$;
- ▶ descrivere uno schedule tramite gli insiemi di lavori delle macchine e calcolare carichi e makespan;
- ▶ contare gli assegnamenti e gli schedule distinti di un'istanza;
- ▶ costruire a mano un primo schedule, assegnando ogni lavoro alla macchina meno carica;
- ▶ calcolare un lower bound sul makespan ottimo e usarlo per dimostrare che uno schedule è ottimo.

Prerequisiti

- ▶ Da questo modulo, il capitolo «**Problemi di ottimizzazione combinatoria**»:
 - ▶ che cos'è un problema di ottimizzazione combinatoria;
 - ▶ il numero di soluzioni candidate;
 - ▶ i bound primali e duali;
 - ▶ le partizioni di un insieme (numeri di Bell).
- ▶ Da questo modulo, il capitolo «**Problemi di packing**»:
 - ▶ il problema del bin packing.

Perché lo scheduling su macchine parallele?

Introduzione: Perché lo scheduling su macchine parallele?

- ▶ Partiamo da una domanda concreta: come ripartire dei lavori fra macchine identiche, in modo che tutti i lavori siano completati il prima possibile
- ▶ La stessa domanda si pone per le stampanti e per i processori di un calcolatore
- ▶ Proponiamo una sfida su una piccola istanza: il capitolo la risolve
- ▶ Ricordiamo il posto famoso di questo problema nella storia degli algoritmi di approssimazione

Sette lavori di stampa, tre stampanti

- ▶ Sette lavori di stampa devono essere inviati a tre stampanti identiche.
- ▶ Ogni lavoro viene eseguito senza interruzioni su una stampante.
- ▶ Come distribuire i lavori, in modo che siano tutti completati il prima possibile?
- ▶ La stessa domanda si pone per i compiti assegnati ai processori di un calcolatore.

Una sfida

L'istanza di riferimento del capitolo: $m = 3$ macchine e $n = 7$ lavori, con tempi di processamento

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$	$j = 7$
t_j	5	5	4	4	3	3	3

- ▶ Prima di proseguire, proviamo a trovare un assegnamento dei lavori che finisca all'istante 9.
- ▶ Poi chiediamoci perché nessun assegnamento può finire prima.
- ▶ Il capitolo risponde a entrambe le domande.

Scheduling e algoritmi di approssimazione

- ▶ Lo scheduling su macchine identiche ha un posto famoso nella storia degli algoritmi di approssimazione.
- ▶ Nel 1966 R. L. Graham ha dimostrato che qualunque lista di lavori, processata in modo greedy, dà un makespan al più $2 - 1/m$ volte quello ottimo.
- ▶ Nel 1969 ha dimostrato il rapporto migliore $4/3 - 1/(3m)$ per la lista ordinata per tempo di processamento non crescente (la regola detta LPT, annunciata nell'articolo del 1966).
- ▶ Queste garanzie, e gli algoritmi che le realizzano, si studiano più avanti nel corso, con gli algoritmi euristici e di approssimazione; questo capitolo introduce il problema e il suo bound più semplice.

Riepilogo

La domanda:

- ▶ Ripartire dei lavori fra macchine identiche (stampanti, processori) in modo che siano tutti completati il prima possibile

La sfida:

- ▶ Completare l'istanza di riferimento all'istante 9, e spiegare perché nessun assegnamento finisce prima

La storia:

- ▶ I rapporti di Graham $2 - 1/m$ (1966) e $4/3 - 1/(3m)$ (1969, LPT): più avanti nel corso

Panoramica della famiglia

Introduzione: Panoramica della famiglia

- ▶ In questa sezione descriviamo che cosa hanno in comune i problemi di scheduling del capitolo: i dati, le soluzioni e l'obiettivo
- ▶ I membri della famiglia differiscono solo per l'ambiente delle macchine
- ▶ La notazione a tre campi è il modo standard di dare un nome a un problema di scheduling: dà al problema del capitolo il suo nome, $P \parallel C_{\max}$

Dati, soluzioni e obiettivo

- ▶ I **problemi di scheduling** di questo capitolo hanno gli stessi dati: un insieme $\mathcal{J} = \{1, 2, \dots, n\}$ di n lavori, m macchine parallele e, per ogni lavoro, un *tempo di processamento*.
- ▶ Una soluzione assegna ogni lavoro a una macchina: i lavori J_h della macchina h sono processati uno dopo l'altro, e la macchina li completa all'istante L_h , il suo **carico**.
- ▶ L'obiettivo è minimizzare il **makespan** $C_{\max} = \max_h L_h$, l'istante di completamento dell'ultimo lavoro.

Lo scheduling su macchine parallele in breve

Un problema di scheduling su macchine parallele ripartisce i lavori fra le macchine in modo che l'ultima macchina finisca il prima possibile.

I membri della famiglia differiscono per l'ambiente delle macchine, descritto dalla notazione a tre campi.

I membri della famiglia

- La tabella elenca i membri della famiglia.

Problema	Notazione	Macchine
macchine parallele identiche	$P \parallel C_{\max}$	ogni lavoro richiede t_j su ogni macchina
macchine con velocità diverse	$Q \parallel C_{\max}$	la macchina h ha una propria velocità
macchine parallele scorrelate	$R \parallel C_{\max}$	il lavoro j richiede t_{jh} sulla macchina h

- $P \parallel C_{\max}$ è definito nella sezione «Scheduling su macchine parallele identiche»; $Q \parallel C_{\max}$ e $R \parallel C_{\max}$ sono descritti dalla notazione a tre campi.

La notazione a tre campi

- ▶ I problemi di scheduling si classificano con la **notazione a tre campi** $\alpha | \beta | \gamma$ di Graham, Lawler, Lenstra e Rinnooy Kan, dove:
 1. α descrive l'**ambiente delle macchine**: 1 per una sola macchina, P per macchine parallele identiche, Q per macchine parallele con velocità diverse, R per macchine parallele scorrelate;
 2. β descrive le **caratteristiche dei lavori**, come le date di rilascio r_j o i vincoli di precedenza $prec$; un campo vuoto indica che non ce ne sono;
 3. γ descrive la **funzione obiettivo**, come il makespan $C_{\max}$ o la somma dei tempi di completamento $\sum_j C_j$.
- ▶ In questa notazione, $P || C_{\max}$ è il problema di schedulare lavori senza caratteristiche aggiuntive su macchine parallele identiche, minimizzando il makespan.

Riepilogo

Struttura comune:

- ▶ Dati: n lavori, m macchine parallele, un tempo di processamento per ogni lavoro
- ▶ Soluzione: ogni lavoro assegnato a una macchina; il carico L_h è l'istante di completamento della macchina h
- ▶ Obiettivo: minimizzare il makespan $C_{\max} = \max_h L_h$

Notazione a tre campi $\alpha | \beta | \gamma$:

- ▶ Ambiente delle macchine $(1, P, Q, R)$, caratteristiche dei lavori, funzione obiettivo
- ▶ $P || C_{\max}$: macchine parallele identiche, nessuna caratteristica aggiuntiva, makespan

Scheduling su macchine parallele identiche



Introduzione: Scheduling su macchine parallele identiche

- ▶ In questa sezione definiamo il problema $P || C_{\max}$
- ▶ Vediamo come si descrivono un'istanza e una soluzione: il vettore degli assegnamenti, gli insiemi di lavori delle macchine, i carichi e il makespan
- ▶ Introduciamo l'istanza di riferimento del capitolo, usata in tutte le sezioni successive
- ▶ Una definizione precisa è ciò che serve per contare le soluzioni, costruire uno schedule e calcolare un bound

Definizione 1: Scheduling su macchine parallele identiche

Dati:

1. *un insieme di m macchine identiche,*
2. *un insieme $\mathcal{J} = \{1, 2, \dots, n\}$ di n lavori,*
3. *per ogni lavoro $j \in \mathcal{J}$, un tempo di processamento $t_j \in \mathbb{Z}_{\geq 1}$,*

il problema $P \parallel C_{\max}$ chiede di assegnare ogni lavoro a una macchina, in modo da minimizzare il tempo necessario per completare tutti i lavori, cioè il massimo tempo di processamento totale assegnato a una macchina.

Il nome e l'istanza

- ▶ Il nome $P \parallel C_{\max}$ deriva dalla notazione a tre campi dei problemi di scheduling.
- ▶ Un'istanza di $P \parallel C_{\max}$ è specificata da:
 1. due numeri interi:
 - il numero di macchine $m \in \mathbb{Z}_{\geq 1}$,
 - il numero di lavori $n \in \mathbb{Z}_{\geq 1}$,
 2. un vettore di numeri interi di lunghezza n :
 - $t \in \mathbb{Z}_{\geq 1}^n$, che contiene i tempi di processamento.

Soluzioni ammissibili

- Una **soluzione ammissibile di** $P \parallel C_{\max}$ è un assegnamento dei lavori alle macchine, che memorizziamo in un vettore $\text{mach} = (\text{mach}_1, \text{mach}_2, \dots, \text{mach}_n)$:

$\text{mach}_j = h \iff$ il lavoro j è assegnato alla macchina h .

- L'insieme dei lavori assegnati alla macchina h è

$$J_h = \{j \in \mathcal{J} : \text{mach}_j = h\},$$

e il suo **carico** è

$$L_h = \sum_{j \in J_h} t_j.$$

Carichi e makespan

- ▶ Ogni macchina processa i suoi lavori uno dopo l'altro, senza tempi morti, quindi la macchina h completa i suoi lavori all'istante L_h .
- ▶ Il **makespan** della soluzione è l'istante di completamento dell'ultimo lavoro,

$$C_{\max} = \max_{h \in \{1, 2, \dots, m\}} L_h.$$

- ▶ L'ordine in cui una macchina processa i suoi lavori non cambia il suo carico: una soluzione è descritta completamente dagli insiemi $J_1, J_2, \dots, J_m$, che formano una partizione di $\mathcal{J}$ (alcuni di essi possono essere vuoti).

Il problema in formule

Data un'istanza di $P \parallel C_{\max}$, il problema si può enunciare formalmente come:

$$\min \left\{ \max_{h \in \{1, \dots, m\}} \sum_{j \in J_h} t_j : J_1, J_2, \dots, J_m \text{ è una partizione di } \mathcal{J} \right\}.$$

- Indichiamo con $z(I)$ il makespan ottimo dell'istanza I .

Ipotesi

► Facciamo la seguente ipotesi:

1. Ci sono più lavori che macchine (altrimenti ogni lavoro ha una macchina tutta per sé, e il makespan ottimo è $\max_j t_j$).

Formalmente:

$$n > m$$

Dalla definizione a un'istanza

- ▶ Fissiamo ora una piccola istanza, con 3 macchine e 7 lavori.
- ▶ È l'**istanza di riferimento** del capitolo: su di essa conteremo gli schedule, costruiremo a mano un primo schedule e dimostreremo che uno schedule è ottimo.

Esempio: l'istanza di riferimento (1/2)

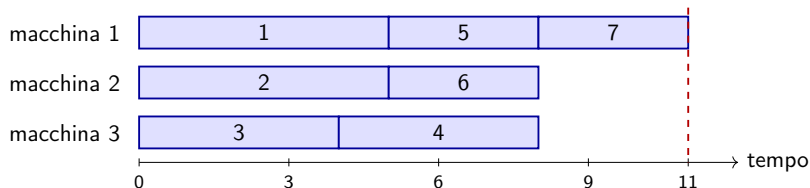
Consideriamo $m = 3$ macchine e un insieme $\mathcal{J} = \{1, 2, \dots, 7\}$ di 7 lavori ($n = 7$), con i seguenti tempi di processamento:

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$	$j = 7$
t_j	5	5	4	4	3	3	3

- Chiamiamo questa istanza **l'istanza di riferimento** del capitolo.

Esempio: l'istanza di riferimento (2/2)

L'assegnamento $\text{mach} = (1, 2, 3, 3, 1, 2, 1)$ dà $J_1 = \{1, 5, 7\}$,
 $J_2 = \{2, 6\}$, $J_3 = \{3, 4\}$, con carichi $L_1 = 11$, $L_2 = 8$, $L_3 = 8$ e
makespan 11:



Riepilogo

Il problema $P \parallel C_{\max}$:

- ▶ Assegnare ogni lavoro a una di m macchine identiche, minimizzando il carico massimo
- ▶ Istanza: m , n e il vettore $\mathbf{t}$ dei tempi di processamento; ipotesi $n > m$

Soluzioni:

- ▶ Il vettore mach , oppure la partizione $J_1, J_2, \dots, J_m$ dei lavori
- ▶ Carichi $L_h = \sum_{j \in J_h} t_j$, makespan $C_{\max} = \max_h L_h$; $z(I)$ è il makespan ottimo

Istanza di riferimento:

- ▶ $m = 3$, $\mathbf{t} = (5, 5, 4, 4, 3, 3, 3)$; l'assegnamento $(1, 2, 3, 3, 1, 2, 1)$ ha makespan 11

Contare le soluzioni



Introduzione: Contare le soluzioni

- ▶ In questa sezione contiamo le soluzioni di un'istanza di $P \parallel C_{\max}$
- ▶ Distinguiamo gli assegnamenti dagli schedule distinti: con macchine identiche, molti assegnamenti descrivono lo stesso schedule
- ▶ Il conteggio mostra perché elencare tutte le soluzioni è senza speranza già per istanze piccole

Assegnamenti e schedule distinti

- ▶ Ogni lavoro può essere assegnato a una qualunque delle m macchine, indipendentemente dagli altri: ci sono m^n assegnamenti mach.
- ▶ Le macchine sono identiche, quindi due assegnamenti che differiscono solo per i nomi delle macchine descrivono lo stesso schedule.
- ▶ Gli schedule distinti sono le partizioni di $\mathcal{J}$ in al più m insiemi non vuoti, e il loro numero è

$$\sum_{k=1}^m S(n, k),$$

dove $S(n, k)$ è il numero di Stirling di seconda specie, cioè il numero di partizioni di n elementi in esattamente k insiemi non vuoti.

Esempio: contare gli schedule (1/2)

- ▶ Per l'istanza di riferimento ci sono $3^7 = 2187$ assegnamenti.
- ▶ I numeri di Stirling sono $S(7, 1) = 1$, $S(7, 2) = 63$ e $S(7, 3) = 301$, quindi gli schedule distinti sono

$$1 + 63 + 301 = 365.$$

Esempio: contare gli schedule (2/2)

- Per enumerazione, gli schedule con il makespan minimo, 9, sono esattamente due:

$$\{1, 3\}, \{2, 4\}, \{5, 6, 7\} \quad \text{e} \quad \{1, 4\}, \{2, 3\}, \{5, 6, 7\},$$

che corrispondono a $3! \cdot 2 = 12$ dei 2187 assegnamenti.

- Lo schedule ottimo quindi non è unico.

Crescita esponenziale

- La crescita è esponenziale in n : con 3 macchine, 20 lavori hanno $3^{20} \approx 3.5 \cdot 10^9$ assegnamenti, e 30 lavori ne hanno $3^{30} \approx 2.1 \cdot 10^{14}$.

macchine m	lavori n	assegnamenti m^n
3	7	$3^7 = 2187$
3	20	$3^{20} \approx 3.5 \cdot 10^9$
3	30	$3^{30} \approx 2.1 \cdot 10^{14}$

Riepilogo

Assegnamenti:

- ▶ m^n vettori mach: ogni lavoro può andare su qualunque macchina

Schedule distinti:

- ▶ Partizioni dei lavori in al più m insiemi non vuoti:
$$\sum_{k=1}^m S(n, k)$$
- ▶ Istanza di riferimento: 2187 assegnamenti, 365 schedule, due schedule ottimi con makespan 9

Crescita:

- ▶ Esponenziale in n : elencare tutte le soluzioni non è un metodo praticabile

Un primo schedule a mano

Introduzione: Un primo schedule a mano

- ▶ In questa sezione costruiamo uno schedule a mano, con una regola semplice: ogni lavoro va alla macchina meno carica
- ▶ La regola è rapida e dà uno schedule ammissibile, il cui makespan è un upper bound su quello ottimo
- ▶ La applichiamo all'istanza di riferimento, e ci chiediamo quanto il risultato sia lontano dall'ottimo

La macchina meno carica

Un modo semplice per costruire uno schedule:

- ▶ prendere i lavori uno alla volta, nell'ordine $1, 2, \dots, n$;
- ▶ assegnare ogni lavoro alla macchina che in quel momento ha il carico minimo;
- ▶ in caso di pareggio, prendere la macchina con l'indice minimo.

Esempio: la macchina meno carica

Sull'istanza di riferimento i carichi delle tre macchine evolvono come segue:

lavoro j (t_j)	macchina scelta	carichi (L_1, L_2, L_3)
1 (5)	1	(5, 0, 0)
2 (5)	2	(5, 5, 0)
3 (4)	3	(5, 5, 4)
4 (4)	3	(5, 5, 8)
5 (3)	1	(8, 5, 8)
6 (3)	2	(8, 8, 8)
7 (3)	1	(11, 8, 8)

Otteniamo l'assegnamento $\text{mach} = (1, 2, 3, 3, 1, 2, 1)$ dell'istanza di riferimento, con makespan 11.

Quanto è buono questo schedule?

- ▶ Quando arriva l'ultimo lavoro tutte le macchine hanno carico 8: qualunque macchina lo riceva, il carico massimo diventa $8 + 3 = 11$.
- ▶ Non sappiamo ancora se 11 è lontano dal makespan ottimo: ci serve un lower bound.

La regola:

- ▶ Lavori nell'ordine $1, 2, \dots, n$; ognuno alla macchina meno carica; i pareggi all'indice minimo

Sull'istanza di riferimento:

- ▶ Assegnamento $(1, 2, 3, 3, 1, 2, 1)$, makespan 11
- ▶ Il makespan di uno schedule è un upper bound: per giudicarlo serve un lower bound

Un lower bound combinatorio

Introduzione: Un lower bound combinatorio

- ▶ In questa sezione ricaviamo un lower bound sul makespan ottimo da due fatti semplici: il lavoro più lungo e il carico medio
- ▶ Il makespan di uno schedule è un upper bound su quello ottimo; il lower bound dice che nessuno schedule può finire prima
- ▶ Quando i due valori coincidono, lo schedule è ottimo: usiamo il bound per risolvere l'istanza di riferimento

Osservazione 1

Per ogni istanza di $P \parallel C_{\max}$, il makespan ottimo $z(I)$ soddisfa

$$z(I) \geq \max \left\{ \max_{j \in \mathcal{J}} t_j, \left\lceil \frac{\sum_{j \in \mathcal{J}} t_j}{m} \right\rceil \right\}.$$

Perché il bound vale

- ▶ La macchina che processa il lavoro più lungo ha carico almeno $\max_j t_j$.
- ▶ Gli m carichi sommano a $\sum_j t_j$, quindi il maggiore è almeno pari alla loro media

$$\frac{1}{m} \sum_j t_j$$

e, essendo intero, almeno pari al suo arrotondamento per eccesso.

Dal bound a una dimostrazione di ottimalità

- ▶ Uno schedule dà un upper bound su $z(I)$: il suo makespan.
- ▶ L'osservazione dà un lower bound su $z(I)$.
- ▶ Se uno schedule ha makespan uguale al lower bound, nessuno schedule può fare meglio: è ottimo.

Esempio: il bound sull'istanza di riferimento (1/2)

- Sull'istanza di riferimento il bound è

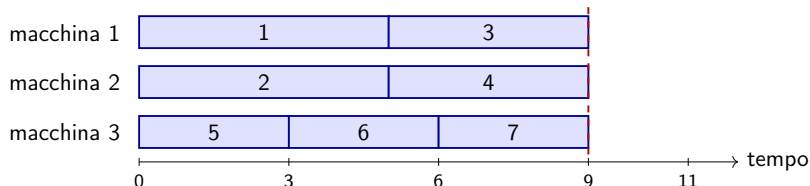
$$\max\{5, \lceil 27/3 \rceil\} = 9.$$

- Insieme allo schedule costruito con la macchina meno carica,

$$9 \leq z(I) \leq 11.$$

Esempio: il bound sull'istanza di riferimento (2/2)

L'assegnamento $\text{mach} = (1, 2, 1, 2, 3, 3, 3)$ dà $J_1 = \{1, 3\}$, $J_2 = \{2, 4\}$, $J_3 = \{5, 6, 7\}$, tutti con carico 9, e makespan 9:



Raggiunge il lower bound, quindi è ottimo: $z(I) = 9$. Le tre macchine finiscono insieme, e non resta alcun tempo morto prima del makespan.

Il bound non è sempre raggiunto

- ▶ Consideriamo $m = 2$ macchine e tre lavori con tempi di processamento $(4, 3, 3)$.
- ▶ Il bound dà $\max\{4, \lceil 10/2 \rceil\} = 5$.
- ▶ Lo schedule migliore è $\{1\}, \{2, 3\}$, con makespan 6.

Il lower bound:

- ▶ $z(I) \geq \max \left\{ \max_j t_j, \left\lceil \frac{1}{m} \sum_j t_j \right\rceil \right\}$: lavoro più lungo e carico medio arrotondato

Ottimalità:

- ▶ Istanza di riferimento: $9 \leq z(I) \leq 11$, poi lo schedule $\{1, 3\}, \{2, 4\}, \{5, 6, 7\}$ raggiunge 9, quindi $z(I) = 9$
- ▶ Il bound non è sempre raggiunto: $(4, 3, 3)$ su due macchine dà 5, l'ottimo è 6

Relazione con il problema del bin packing

Introduzione: Relazione con il problema del bin packing

- ▶ In questa sezione confrontiamo $P || C_{\max}$ con il problema del bin packing (BPP)
- ▶ I due problemi pongono la stessa domanda da lati opposti: le macchine sono i contenitori, i carichi sono i pesi totali
- ▶ Per questo idee e algoritmi per uno dei due hanno spesso un corrispondente per l'altro

Schedule e riempimenti

- ▶ Dato un valore T , esiste una schedule con makespan al più T se e solo se i lavori, visti come oggetti di peso t_j , possono essere inseriti in m contenitori di capacità T .
- ▶ Le macchine sono i contenitori e i carichi sono i pesi totali dei pattern di riempimento.

Quantità opposte

- ▶ I due problemi fissano e minimizzano quantità opposte:

	numero di contenitori / macchine	capacità / makespan
BPP	minimizzato	fissato (c)
$P \parallel C_{\max}$	fissato (m)	minimizzato

- ▶ Per questo i due problemi condividono molte idee: diversi algoritmi euristici per uno dei due hanno un corrispondente per l'altro.

Lo scheduling come packing:

- ▶ Makespan al più $T \iff$ i lavori entrano in m contenitori di capacità T

Problemi speculari:

- ▶ BPP: capacità fissata, si minimizza il numero di contenitori
- ▶ $P \parallel C_{\max}$: numero di macchine fissato, si minimizza il makespan
- ▶ Gli algoritmi euristici per uno dei due hanno spesso un corrispondente per l'altro

Osservazioni

Introduzione: Osservazioni

- ▶ Chiudiamo il capitolo con due risultati classici di R. L. Graham sullo scheduling greedy di una lista di lavori
- ▶ Dicono quanto uno schedule greedy può essere lontano da quello ottimo, nel caso peggiore
- ▶ Gli algoritmi corrispondenti si studiano più avanti nel corso



I risultati di Graham

- ▶ Nel 1966 R. L. Graham ha dimostrato che qualunque lista di lavori, processata in modo greedy, dà un makespan al più $2 - 1/m$ volte quello ottimo.
- ▶ Nel 1969 ha dimostrato il rapporto migliore $4/3 - 1/(3m)$ per la lista ordinata per tempo di processamento non crescente (la regola detta LPT).

Scheduling greedy di una lista:

- ▶ Qualunque lista: makespan al più $2 - 1/m$ volte quello ottimo (1966)
- ▶ Lista ordinata per tempo di processamento non crescente (LPT): al più $4/3 - 1/(3m)$ volte quello ottimo (1969)

Errori tipici

Introduzione: Errori tipici

- ▶ In questa sezione raccogliamo gli errori più comuni su $P \parallel C_{\max}$
- ▶ Ogni errore nasconde un'idea chiave del capitolo: macchine identiche, makespan, interezza, bound duali e ottimi multipli
- ▶ Riconoscerli aiuta a leggere e risolvere correttamente gli esercizi

Osservazione 2: Attenzione

- ▶ **Contare m^n schedule diversi.** Le macchine sono identiche: rinominarle dà lo stesso schedule, quindi gli schedule distinti sono le partizioni dei lavori in al più m insiemi.
- ▶ **Confondere makespan e carico totale.** Il makespan è il carico massimo di una macchina, non la somma dei tempi di processamento e non il carico medio.
- ▶ **Dimenticare l'arrotondamento per eccesso.** I tempi di processamento sono interi, e quindi anche il makespan: il carico medio $27/3$ si arrotonda per eccesso quando non è intero.

Osservazione 3: Attenzione

- ▶ *Supporre che il bound sia raggiunto. Il lower bound combinatorio è un bound duale: una schedule con quel makespan può non esistere, come mostra l'istanza $(4, 3, 3)$ su due macchine.*
- ▶ *Credere che lo schedule ottimo sia unico. L'istanza di riferimento ha due schedule ottimi e dodici assegnamenti ottimi.*

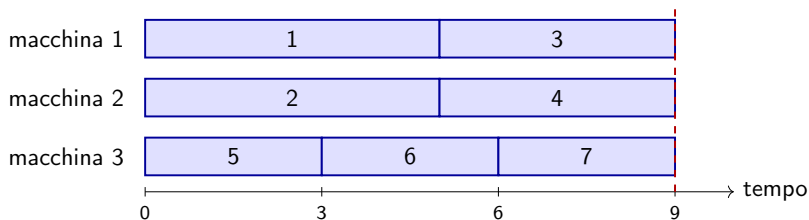
Schedule e makespan:

- ▶ Gli schedule distinti sono partizioni, non assegnamenti
- ▶ Il makespan è il carico massimo, un intero: il carico medio si arrotonda per eccesso

Bound e ottimi:

- ▶ Un bound duale può non essere raggiunto
- ▶ Lo schedule ottimo può non essere unico

La sfida, risolta



- ▶ $J_1 = \{1, 3\}$, $J_2 = \{2, 4\}$, $J_3 = \{5, 6, 7\}$: tutti i lavori sono completati all'istante 9.
- ▶ Nessun assegnamento finisce prima: 9 è il lower bound $\max\{5, \lceil 27/3 \rceil\}$.

Riepilogo del capitolo (1/3)

- ▶ Il problema $P \parallel C_{\max}$ chiede di assegnare lavori a macchine identiche minimizzando il carico massimo, il makespan; una soluzione è una partizione dei lavori in al più m insiemi.
- ▶ Ci sono m^n assegnamenti, ma solo $\sum_{k=1}^m S(n, k)$ schedule distinti.

Riepilogo del capitolo (2/3)

- ▶ Dare ogni lavoro alla macchina meno carica costruisce rapidamente uno schedule; il massimo fra il lavoro più lungo e il carico medio arrotondato è un lower bound sul makespan ottimo.
- ▶ Quando uno schedule raggiunge il lower bound è ottimo; altrimenti il gap resta aperto.
- ▶ Il problema è lo speculare del bin packing: qui il numero di macchine è fissato e la capacità è minimizzata.

Riepilogo del capitolo (3/3)

Quadro dei bound dell'istanza di $P \parallel C_{\max}$ (minimizzazione: il lower bound è un bound duale, l'upper bound è il makespan di uno schedule):

Capitolo	Lower bound	Upper bound	Da
Questo capitolo	9	11	lavoro più lungo e carico medio; macchina meno carica
Questo capitolo, schedule migliorato	9	9	$J_1 = \{1, 3\}$, $J_2 = \{2, 4\}$, $J_3 = \{5, 6, 7\}$