

UNIVERSITÀ DEGLI STUDI DI MODENA E
REGGIO EMILIA

Dipartimento di Scienze Fisiche, Informatiche e
Matematiche

CORSO: ALGORITMI DI OTTIMIZZAZIONE

Schedulazione lavori su macchine industriali

CANDIDATO: ANGELO LONGANO

PROFESSORE:
PROF. MAURO DELL'AMICO

Anno Accademico 2023/2024

1 Problema aziendale

1.1 Descrizione generale

L'azienda AP srl di Reggio Emilia è un'**azienda manifatturiera** specializzata nel settore metallurgico.

L'impresa si occupa di lavorazione, trattamento e rifinitura di metalli semilavorati in metallo e leghe per conto terzi.

L'azienda vuole migliorare l'efficienza dei suoi **processi produttivi** per riuscire ad essere più competitiva nel mercato. I loro processi partono da un prodotto grezzo che deve essere lavorato da una delle macchine presenti in azienda.

L'azienda ha un magazzino interno per ospitare prodotti grezzi e prodotti finiti, in questo studio non viene considerata una capienza massima.

Nel problema viene considerato un periodo di N giorni, dove i prodotti grezzi sono già presenti in magazzino, e al termine del periodo bisogna soddisfare scadenze dei cliente con le quantità di prodotto finito desiderate.

Obiettivo L'azienda vuole una **schedulazione** di N giorni dove vengono indicate le lavorazioni da assegnare alle macchine e il numero di prodotti da ottenere.

La schedulazione deve rispettare i vincoli del problema e soprattutto raggiungere le quantità richieste dai clienti.

La soluzione deve ridurre i costi legati al cambio configurazione e all'utilizzo superfluo di macchine, cercando di ottenere idealmente solo macchine che lavorano.

1.2 Analisi processo produttivo

Il processo è formato da **diverse fasi** (come si può notare anche in figura 1):

1. Configurazione macchina (tempo costante).
2. L'operatore posiziona il prodotto grezzo in uno dei pallet dedicato della macchina. (tempo costante)
3. La macchina effettua la lavorazione. (tempo dinamico, dipende dal prodotto)

4. L'operatore controlla il prodotto lavorato, effettua i vari test qualità e sposta il prodotto finito fuori dalla macchina (tempo costante).
5. Pulizia macchina (tempo costante).

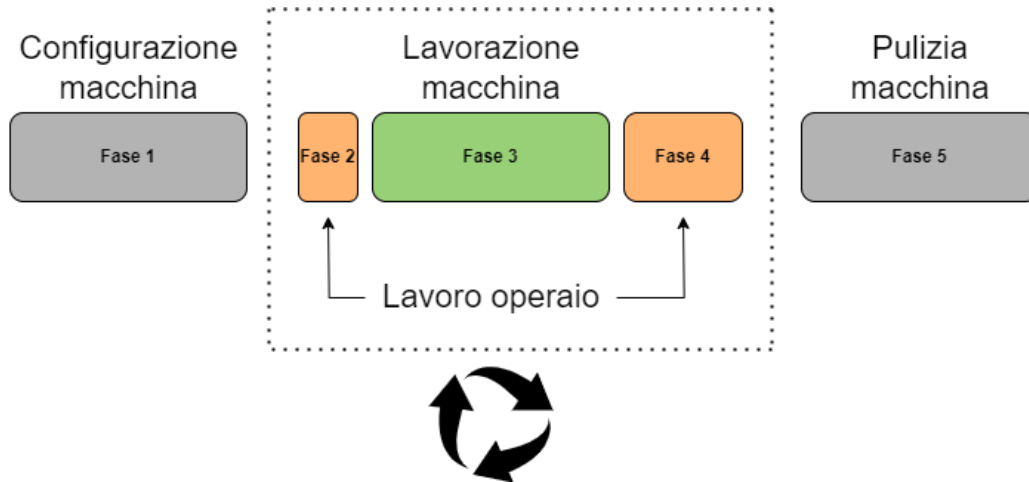


Figura 1: Fasi di produzione

Per ottenere un prodotto è necessario disporre il prodotto grezzo, aspettare la lavorazione della macchina e controllare il prodotto finito.

La prima e l'ultima fase sono presenti una sola volta agli estremi dell'intero processo produttivo, mentre le fasi 2, 3 e 4 vengono ripetute ciclicamente per raggiungere la quantità di prodotto desiderata.

Quindi per produrre 100 prodotti, sarà necessario configurare la macchina, effettuare 100 cicli di lavorazione e pulire la macchina.

In questo modo si può creare una pipeline di produzione come si può notare in figura 2, e parallelizzare il lavoro della macchina con quello dell'operatore.

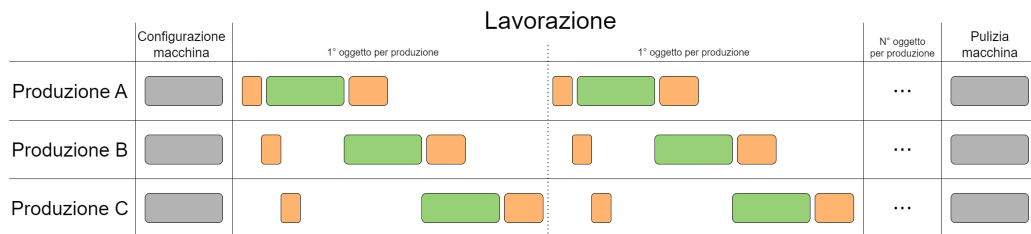


Figura 2: Pipeline di produzione su una macchina

L'operatore posiziona il primo pezzo, fa partire la lavorazione e nel mentre posiziona l'altro pezzo da lavorare.

Allo stesso modo l'operatore può controllare il prodotto finito nel mentre che la macchina lavora su un altro pezzo.

Analizzando un ciclo della lavorazione in pipeline (in figura 3), il tempo di lavorazione dell'operaio si paga **per un solo prodotto** rispetto agli N prodotti: fase 2 primo prodotto e fase 4 ultimo prodotto;

mentre il restante tempo necessario alla lavorazione dell'operaio è coperto dalla lavorazione della macchina. Questo porta ad un risparmio di tempo, ma si ha un ciclo di produzione più lungo: ottenere 1 prodotto finito di ogni N prodotti, è necessario aspettare la lavorazione di tutti gli N prodotti.

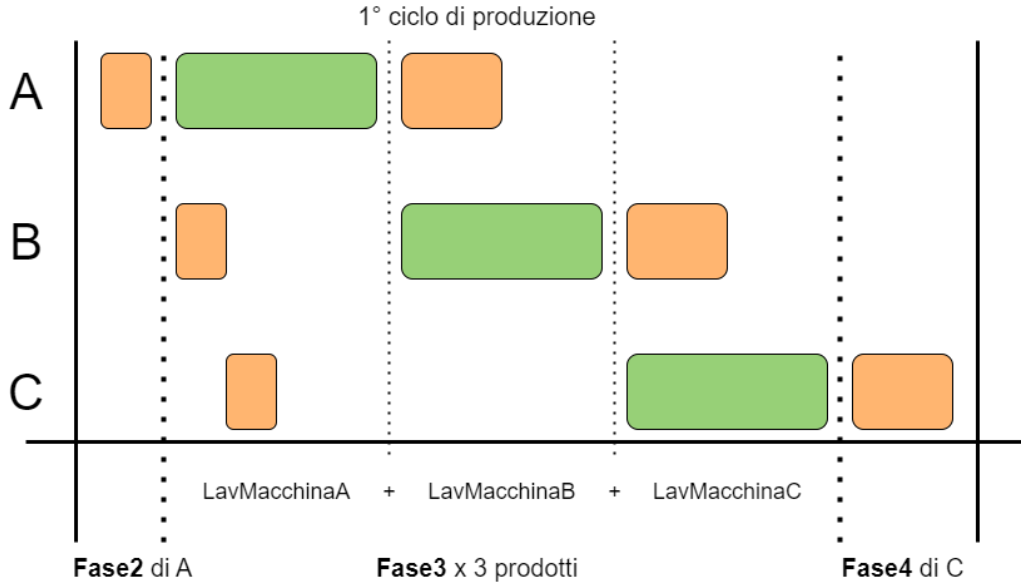


Figura 3: Analisi singolo ciclo di produzione

I tempi delle diverse fasi di produzioni possono essere semplificate come segue:

1. *TIME_MACHINE_SETUP* : tempo configurazione e pulizia macchina (fase 1 e fase 5).
2. *TIME_OPERATOR* : tempo di lavorazione manuale (fase 2 e fase 4).
3. *time_machine_w* : tempo di lavorazione automatica della macchina (fase 3).

I primi due sono tempi costanti, mentre il tempo di lavorazione della macchina dipende dal prodotto da lavorare.

L'algoritmo dovrà scegliere quanti e quali lavorazioni fare in una determinata macchina e quale quantità di prodotto raggiungere.

Queste scelte vengono definite come **Jobs** e sono formati da una lista di lavorazioni (*listWorks* e prodotti da raggiungere *WorkedQuantity*).

Per definire un lavorazione sequenziale, l'algoritmo creerà diversi jobs con un solo elemento nella lista di lavorazioni.

Mentre per lavorare in pipeline N lavorazioni, basta creare un solo job.

In conclusione la **differenza** tra effettuare N lavorazioni sequenzialmente (su

N jobs) o in una pipeline per raggiungere P prodotti è la seguente:

Lavorazione **sequenziale** con N jobs

$$TotalTime = N \times TIME_MACHINE_SETUP + \left(\sum_w^{Works} time_machine_w + \sum_w^{Works} TIME_OPERATOR \right) \times P \quad (1)$$

$$TotalCost = N \quad (2)$$

Lavorazione in **pipeline**

$$TotalTime = TIME_MACHINE_SETUP + \left(\sum_w^{Works} time_machine_w + TIME_OPERATOR \right) \times P \quad (3)$$

$$TotalCost = 1 \quad (4)$$

1.3 Formalizzazione del problema

Sono date M **macchine** (*Machines*) con una velocità e costo di produzione identico, ma ognuna ha un numero di pallet ($pallets_m$, $m \in M$) che indica il numero di lavorazioni in parallelo.

Sono dati W **tipologie di lavorazione** (*TypeWorks*) che hanno un tempo macchina necessario alla loro lavorazione ($timeMachineNeeded_w$, $w \in W$). Ad ogni lavorazione è associato in modo implicito il prodotto grezzo usato e il prodotto finito lavorato.

Ad ogni macchina bisogna associare dei N Jobs che hanno una lista di lavorazioni (*listWorks*) e una quantità da raggiungere per ogni lavorazione (*WorkedQuantity*).

I Jobs sono creati dinamicamente dall'algoritmo e hanno un *timeNeeded()* variabile in base alla *listWorks* scelta e la quantità da raggiungere *WorkedQuantity*. Con i jobs si riesce a creare un processo a lavorazione sequenziale (*listWorks*

con una sola lavorazione) e in pipeline (*listWorks* con più lavorazioni).
 Il *timeNeeded()* verrà calcolato di conseguenza.

$$TimePerUnit_j = \left(\sum_w^{Works} timeMachineNeeded_w \right) + TIME_OPERATOR \quad (5)$$

$$timeNeeded()_j = TIME_MACHINE_SETUP + TimePerUnit_j \times len(listWorks) \quad (6)$$

I $w \in listWorks$ devono rispettare il seguente vincolo: il tempo per ogni lavorazione deve essere maggiore del tempo necessario al lavoro dell'operatore.

$$timeMachine_w > TIME_OPERATOR \quad (7)$$

Il costo di configurazione di un job ha sempre lo stesso costo, indipendentemente dal numero di lavorazioni.

Il **tempo** viene misurato in minuti(*MIN*) e in giorni (*DAYS*) da 10 ore lavorative.

Sono dati N scadenze da parte dei clienti dell'azienda (*CustomerDeadlines*) che al termine del periodo considerato desiderano una certa *DesiredQuantity_w* per ogni prodotto ($w \in WorkTypes$).

Scelte dell'algoritmo L'algoritmo deve creare i vari Jobs in modo da distribuire le lavorazioni nelle macchine nel modo più efficiente.

Bisogna lavorare il numero di prodotti desiderati dai clienti entro il termine dei giorni stabilito (N DAYS).

L'algoritmo deve favorire l'utilizzo delle lavorazioni in pipeline rispetto alle sequenziali.

Una buona soluzione minimizza il numero di jobs da creare e minimizza la capacità residua delle macchine (utilizzando meno macchine).