NUMERO: 719 DATA: 07/10/2013

A P P U N T I

STUDENTE: Costantino

MATERIA: Programmazione e Controllo della Produzione
+ Esercizi + temi + casi Prof. Alfieri

PROGRAMMAZIONE e CONTROLLO della PRODUZIONE

Cap 7: The Process View of the Organization

Principale obiettivo di un manager: Fare incontrare domanda e offerta

eccesso di domanda → profitto perso
" " offerta → eccesso di costo

Far incontrare domanda e offerta sarebbe molto semplice se i processi fossero istantanei e si potesse quindi creare immediatamente qualunque quantità di offerta x soddisfare la domanda

↳ bisogna continuamente cercare di produrre QUANTO, COSA e QUANDO richiesto dal mercato

↳ inoltre la capacità produttiva deve essere sufficiente non solo in media ma anche nello specifico istante (es. ospedali Gennaio)

Esistono diversi tipi di MIS-MATCH tra domanda e offerta:

- di tempistica
- quantitativi
- qualitativi
- ...

Difficoltà nel far incontrare domanda e offerta dovute a:

- variabilità domanda (non sempre prevedibile a priori)
  ↳ se mi dimensiono sul MAX della domanda ho troppa capacità e la capacità costa (inoltre aumenterebbe troppo il MG)
- inflessibilità risorse: non possono essere accese/usate o spente/inusate a seconda della domanda
  la capacità è rigida (e quindi anche l'offerta)

(N.B.) Il PREZZO può attenuare Mis Matching ma NON lo può eliminare!
↳ a volte NON c'è neppure il tempo x adeguare prezzo

Una delle cose più importanti x limitare MIS-MATCH è la Tempestività nel soddisfare la domanda

Tempo d'attesa corrisponde al tempo che trascorre tra quando AVREI POTUTO INIZIARE l'attività e quando L'HO EFFETTIVAMENTE INIZIATA

→ se l'offerta (capacità produttiva) fosse infinita non avrei tempi d'attesa e quindi la durata del mio processo coinciderebbe con il cammino critico

→ Non posso conoscere a priori la durata dei tempi d'attesa, pertanto è possibile rappresentarli nel diagramma di Gantt solo ex post

Esiste il tempo d'attesa perché:

- Le risorse sono sempre limitate (*) (non possono essere infinite)
- I tempi di processo, nel mondo reale, NON SONO MAI DETERMINISTICI (c'è sempre variabilità " " ", no tempi esattam. prestabiliti)
  → es. appuntamenti
  → tempi attesa mi servono x non lasciare inutilizzate le risorse

risorsa deve sempre lavorare xk rappresenta un costo in ogni caso (meglio far aspettare i clienti/pezzi che gli operatori/macchinari)

Dal PV delle risorse:

- se la risorsa è utilizzata al 100% x tt l'intervallo temporale considerato → Risorse sottodimensionate (mismatch quantitativo → capacità insufficiente)
- se vi sono momenti in cui è inutilizzata → Risorse sovradimensionate (costi eccessivi)

Conoscendo il tempo effettivo da quando entra l'input a quando esce l'output e conoscendo la media delle durate delle attività, posso stabilire il tempo d'attesa, dichiarando se un processo è ben dimensionato o meno. Oppure, conoscere il numero medio di flow-unit può essere utile x capire se le risorse sono ben utilizzate e se i "luoghi" sono sufficientemente capienti

Tempo Attesa = tot attività ex-post − tot attività ex ante

→ crescono quando più flow units "competono" per la stessa (*) limitata risorsa (simultaneamente)

- Processo = "scatola nera" che usa risorse (lavoro, capitale) per trasformare gli input in output

- Flow Unit (unità di flusso): elemento che "fluisce" nel processo, ossia elemento che viene trasformato da input in output (item in proc

→ definendo la flow unit appropriata, possiamo quindi valutare il processo basandoci sulle 3 grandezze fondamentali x la misura delle performance di processo:

2

N.B. il flow (o TH) rate (del processo) corrisponde alla capacità della risorsa (attività) più scarsa (ossia con capacità minore)

infatti CAPACITÀ e FLOW RATE hanno lo stesso unità di misura → [unità / time]

1ª definizione

• **COLLO DI BOTTIGLIA (Bottleneck)** = risorsa con capacità minore (non quella più lenta!)

N.B. posso definire collo di bottiglia in funzione delle capacità SOLO SE
- ho un UNICA FLOW UNIT
- ho un flusso UNICO (ossia l'unica flow unit attraversa TUTTE le risorse, o meglio tutte le risorse vedono le stesse unità che passano e lo stesso numero)

Come sono legate WIP, FLOW TIME e FLOW RATE?

Legge di Little: $\boxed{\text{Average Inventory} = \text{Average Flow Rate} \cdot \text{Average Flow Time}}$

→ WIP = TH · CT

sempre valida → non dipende da:
- sequenza con la quale le flow unit sono processate (LIFO, FIFO) → può influenzare il flow time di una particolare flow unit ma non la media
- casualità, variabilità

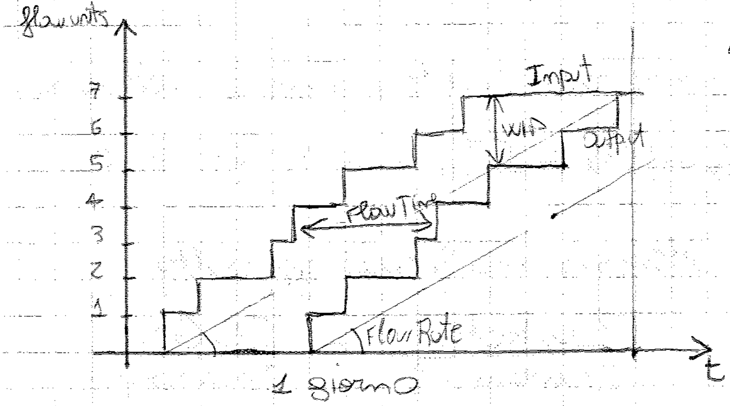
→ utile x trovare una delle 3 misure date le altre 2

→ Non sa dare risposta alle domande:
- xk esiste l'inventory?
- qual è il valore ottimale dell'inventory

questo perché, usando i valori medi, noi abbiamo un'ISTANTANEA della situazione!

x le nostre analisi abbiamo bisogno delle curve cumulate

Nell'esempio del reparto di chirurgia: (OSS. costante → deterministico; deterministico $\not\Rightarrow$ costante)



Assunzioni:
- sequenza NON perturbata (ossia FIFO)
- tempi deterministici costanti (gradini delle stesse ampiezza)
- stessi tempi d'attesa
- arrivi costanti

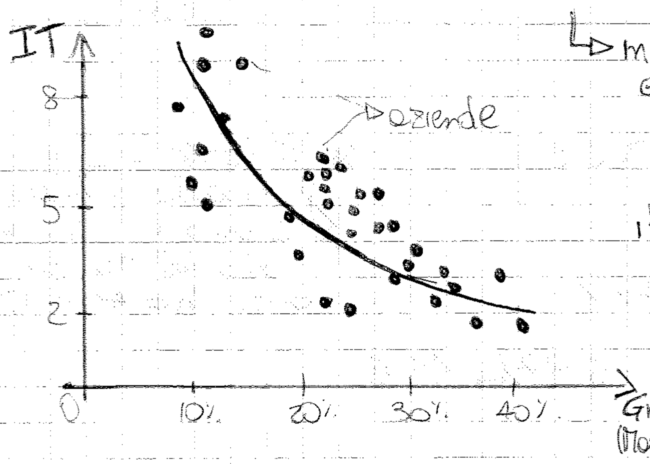
→ se arrivi non fossero costanti i gradini avrebbero ampiezze diverse e le inclinazioni delle rette (flow rate) sarebbero diverse

3

In generale è auspicabile avere un IT alto in modo da smaltire in fretta ciò che ho in MG (riducendo i costi unitari di MG) evitando rischi come obsolescenza ecc... e in modo da convertire velocemente i \$ investiti in guadagni

↓

ma IT dipende anche dal tipo di prodotto o settore di mercato in cui opero



↳ in genere a margini alti corrisp. un IT basso e viceversa (andamento parabolico)

↓

andamento ideale

↓

item molto richiesti e facilm. immettibili nel mercato avranno IT alti e in genere margini bassi

↳ x i beni di lusso ad es. è il contrario

È la strategia aziendale che decide se posizionare l'azienda a dx o a sx di tale curva → nel mondo reale x un certo valore possono corrispondere più valori di IT

- Costo del magazzino: costo finanziario (interessi che avrei ottenuto se avessi investito il capitale immobilizz. in MG)
  - \+ deperibilità e obsolescenza (degli item in MG)
  - \+ furti, perdite, rotture
  - \+ costi operativi MG (affitto, energia, assicuraz. ...)
  - \+ aumento tempi d'attesa
  - \+ qualità (più item restano in MG più tempo impiego ad accorgermi di eventuali errori in produzione)

↓

espresso sottoforma di tasso percentuale

↓

(espresso come costo annuale inventory (i%))

Dividendo il costo annuale di MG per gli IT annuali ottengo il costo unitario di

$$\frac{\text{Costo annuale inventory\%}}{\text{Inventory Turns}} = \frac{\text{Annual \%}}{1/\text{days of supply}} = \frac{\text{Annual \%}}{1/FT} = \frac{\text{Annual \%}}{TH/WIP} =$$

$$= \frac{\text{Annual\%} \cdot WIP}{TH} = \frac{\text{Annual\%} \cdot WIP}{COGS} = \frac{\text{Costo tot annuale MG}}{\text{\$ usciti dal MG (costo del venduto)}} =$$

= costo unitario di MG (per-unit inventory cost) %

↳ quanto costa tenere in MG una singola unità di flusso

4

② Stagionalità ⟶ <u>SEASONAL INVENTORY</u>

occorrono quando la capacità è rigida e la domanda è variabile

es. panettoni: vendo solo in un periodo dell'anno ma, dato che sarebbe troppo costoso aumentare la capacità x questo periodo (capacità che il resto dell'anno sarebbe inutilizzata) e poi smantellarla quando non più necessaria, inizio a produrre parecchi mesi prima e immagazzino → seasonal inventory

Casi simili quando vi è variabilità dell'offerta → es. settore agricolo (pomodori solo d'estate...)

Caratteristiche seasonal inventory:

- Disallineamento temporale tra domanda e offerta
- Legate <u>ANCHE</u> ad una motivazione di costo

③ Economie di Scala ⟶ <u>CYCLE INVENTORY (Scorte Ciclo)</u>

In molte situazioni è economico processare molte flow units contemporaneam. x trarre vantaggi dalle economie di scala

es. TRASPORTI: una spedizione comporta un costo fisso indipendente dalla qtà trasportata. x mitigare tale costo è utile riempire completam. il camion x dividere tale costo fisso tra più unità possibili

necessario un inventory x arrivare alla qtà di carico max

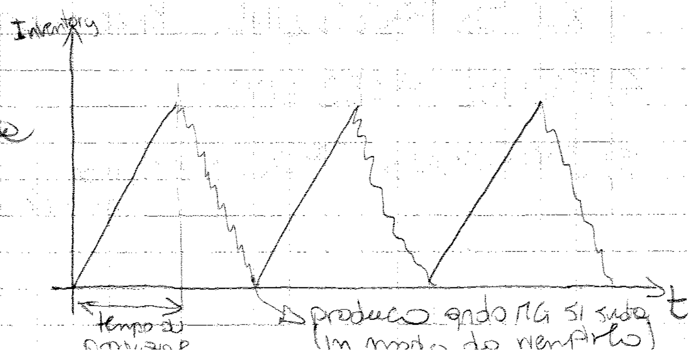
Caratteristiche cycle inventory:

- Non è detto che ci sia disallineamento temporale tra domanda e offerta
- Legate <u>SOLO</u> ad una motivazione di costo

⟹ Seasonal inv. ≠ cycle inventory

→ creato x una motivaz. di costo

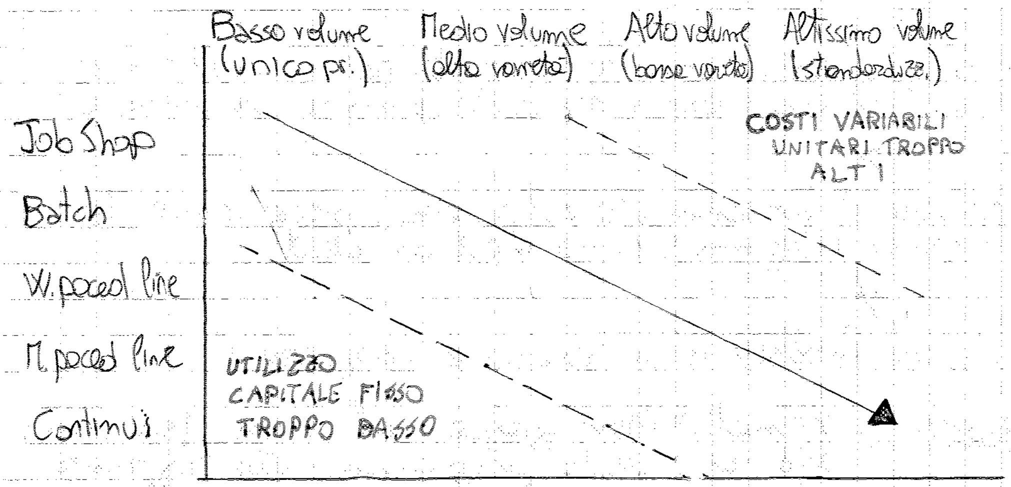
dovuto a squilibri temporali tra domanda e offerta causati da variabilità della domanda o " " offerta



⑤

L'utilità del classificare i processi e dell'utilizzare la matrice prodotto-processo sta soprattutto nel fatto che:

- processi simili tendono ad avere problemi simili
- la naturale tendenza delle industrie verso la parte bassa a destra della matrice ci consente di prevedere come i processi si evolveranno probabilmente in una particolare industria



Col crescere del volume di prod. tendenza a spostarsi lungo la freccia (diagonale)

N.B. Posizionarsi negli angoli in verde non è economico → porta al fallimento

Ricapitolando:

- Molto importante guardare alle operazioni di un'impresa in termini di processo che genera l'offerta, in quanto si notano i mismatch tra domanda e offerta.

  Dalla prospettiva del prodotto tali mismatch assumono la forma di tempi d'attesa; dal P.V. del processo " " " di inventory

- Per ogni processo possiamo definire 3 misure fondamentali della performance:
  - inventory
  - flow time
  - flow rate

  Legate tra loro dalla Legge di Little:

  $$\text{av. inventory} = \text{av. flow time} \cdot \text{av. flow rate}$$

- Una misura correlata al flow time è l'inventory turns:

  $$IT = \frac{1}{\text{flow time}}$$

  che dà informazioni riguardo la velocità con cui le flow units vengono trasformate da input ad output

  questo indicatore sta inoltre alla base del calcolo degli inventory costs associati ad ogni unità

OSS.:
- In genere Buffers sono in grado di gestire flow unit di tipo diverso
- Le Attività invece di solito sono in grado di gestire un unico tipo di flow unit (xk a quel particolare livello di aggregazione non riesco a capire che cosa dovrà fare quella determinata risorsa
- Alcune attività (es. ascensore) non vanno inserite nel Process Flow Diagram xk:
  - non aggiungono valore
  - flow units passano poco tempo in quell'attività (bassi process time)
  - non sono vincolanti x il processo (non lo rallentano, NON sono critiche), cioè non creano attese (se così non fosse dovrei inserirle)
  - ↳ anche se in uno schema sui costi andrebbero inserite
- Nei processi continui NON posso inserire buffers intermedi (devo dimensionare il processo sulla risorsa scarsa)
- Pur non conoscendo nel dettaglio il funzionamento (P.V. ingegneristico) del processo posso cmq analizzarlo (dal P.V. della flow unit)
- Il Process Flow Diagramm NON rappresenta il layout (posizione, dimensione degli elementi NON ha alcun valore)
- Scendendo ad un livello meno aggregato alcune attività diventano a loro volta processi

- **Capacità del processo**: quantità MASSIMA di flow unit che il processo è in grado di produrre/generare nell'unità di tempo

↓

per determinarla devo prima determinare la capacità di ogni singola risorsa:

- Capacità attività/risorsa: qtà MAX di flow unit producibile da una data risorsa nell'unità di tempo

N.B. Capacità ≠ Flow Rate
↳ quanto il processo PUÒ PRODURRE
→ qnto il processo PRODUCE

⇒ Flow Rate ≤ Capacity sempre! (Anche se hanno la stessa unità di misura)

Definizioni:
- capacità processo in serie: capacità del task avente capacità minima
- " " " parallelo: somma delle capacità dei due rami (se nei rami scorre la stessa flow unit) o capacità del ramo con capacità minore (se scorrono flow unità diverse sui 2 rami)

$$\boxed{\text{Capacità processo} = \min_i \left(\text{Capacità attività}_i\right)}$$

↳ Risorsa più stringente (COLLO di BOTTIGLIA)

$$\boxed{\text{Flow Rate processo} = \min\left(\text{capacità proc.}, \text{domanda}, \text{input}\right)}$$

↳ formule valide sotto le ipotesi di domanda costante e assenza di guasti

Utilizzo, così come la capacità, può essere definito a livello di processo o al livello di risorsa (anche se nel primo caso non ha molto senso)

Se processo è Demand-Constrained o Input-constrained nessuna risorsa avrà $U = 100\%$

(N.B.) Nessuna risorsa può avere un utilizzo superiore del collo di bottiglia

Ricorda: nella maggiorparte dei businesses l'obiettivo è MASSIMIZZARE IL PROFITTO, NON L'UTILIZZO!

(N.B.) Utilizzo dà informazioni riguardo l'ECCESSO DI CAPACITÀ, ma da esso non possiamo dedurre quanta domanda eccede la capacità del processo (cioè se domanda > capacità).

Per questo si è definita la:

- Implied utilization (utilizzo implicato) che misura il mismatch tra cosa potrebbe fluire attraverso la risorsa (domanda) e quanto la risorsa può produrre (capacity)

ossia misura la quantità di domanda inevasa a causa della capacità

Dato che l'I.U. è calcolato sulla singola attività/risorsa nella formula viene usato il:

- Workload: domanda vista dall'attività, ossia la domanda che potrebbe attraversare una data risorsa

$$\boxed{\text{Implied Utilization (I-U)} = \frac{\text{Workload (o domanda)}}{\text{Capacity}}}$$

può anche essere ≥ 1

vorrebbe dire che x soddisfare la domanda necessito di più capacità di quella di cui dispongo ora

OSS:

- se I-U > 100% il processo è capacity-constrained
- se una risorsa ha I-U < 100% sicuramente NON è collo di bottiglia (stesso cosa vale per l'utilizzo)
- Qualunque sia il processo considerato non posso affermare con certezza che almeno una risorsa sia utilizzata al 100%, perché se input < capacity e/o processo è demand-constrained NESSUNA RISORSA sarà usata al 100%

↳ posso affermarlo solo se il processo è capacity-constrained

(N.B.) Se domanda < input e domanda < capacità:

⇒ Utilizzo ≡ Utilizzo implied (Demand-constrained)

Ricapitolando: ① Ogni analisi di processo dovrebbe iniziare con la creazione del Process Flow Diagram, specialmente nei casi con flow unit multiple (in quanto i loro flussi sono più complessi)

② Come seconda cosa bisogna identificare il Collo di Bottiglia:

- flow unit e flusso unici: risorsa con capacità minore
- in generale (es. product mix): " " I-U = $\frac{\text{workload risorsa}}{\text{capacity}}$ maggiore

③ Calcolare le varie misure di performance (Flow Rate, ad es, ci permette di calcolare l'utilizzo del processo)

- Analisi di processo con flow unit UNICA:

① Trovare capacità di ogni attività (tenendo conto del fatto che un'attività può necessitare di più risorse)

② l'attività/risorsa con capacità minore è il Collo di bottiglia

Capacità CdB = capacità di processo

③ trovo flow rate = min {input, domanda, capacità}

④ UTILIZZO = $\frac{\text{flow rate}}{\text{capacity}}$

- Analisi di processo con flow units multiple:

① Scegliere flow unit aggregata (es. "minuti di lavoro") e calcolare, x ogni , il numero di minuti impiegati da tale attività

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 19 di 229

$$\boxed{\text{Tempo di uscita di X unità partendo a sistema vuoto} = T_0 + \frac{X-1}{\text{Flow Rate}}}$$, T_0 = Raw Process Time (o Time empty process)

↳ se il sistema è a regime e/o X è grande: Tempo di uscita $= \frac{X}{\text{Flow Rate}}$

OSS. approssimazioni: - X stabilire costi approssimo valori x eccesso
- " " qtà prodotte " " " difetto

↳ politica del worst case (caso peggiore)

- Labor Content: somma dei process times delle attività che coinvolgono lavoro umano (qtà di tempo di lavoro diretto su ogni unità)

$$\boxed{\text{Labor Content} = \sum \text{process time}_i \text{ (delle attività presedute da un umano)}}$$

↳ misura che prende la prospettiva delle flow unit ma non riflette alcune info riguardo a come il processo sta attualmente operando

X questo è sbagliato moltiplicare il Labor Content per la paga oraria dei lavoratori per calcolare il costo del lavoro

bisogna tenere conto anche dei tempi di inattività dei lavoratori che devono cmq essere pagati → bisogna usare Tot stipendi (o costo lavoro totale):

$$\boxed{\text{Costo lavoro diretto per unità} = \frac{\text{Tot. costo lavoro}}{\text{Flow Rate}}}$$

→ attenzione ad usare le stesse unità di tempo e num e denom!! (es. sett)
↳ se paga è 12 €/h e flow rate = 125 unit/sett:
Costo lavoro $= \frac{12 \text{ €/h} \cdot 35 \text{ h/sett} \cdot 3}{125 \text{ unit/sett}}$ ∀ pezzo, n° impiegati

in qst modo considero anche il costo dei tempi di inattività

- Idle Time: tempo non produttivo/di inattività (durante il quale il lavoratore è cmq pagato) di lavoratori o macchine, dovuto ad attese (o in generale ad interruzioni delle operazioni)

$$\boxed{\underset{\text{(della singola risorsa)}}{\text{Idle Time (o Attesa)}} = \text{Cycle Time} - \underset{\text{(singola risorsa)}}{\text{Process Time}}}$$

OSS: se il processo è demand-constrained anche CdB può avere tempi d'attesa

↳ tempo improduttivo di ogni risorsa x unità prodotta

$$\boxed{\underset{\forall \text{ unità}}{\text{Costo attesa}} = \underset{\forall \text{ pezzo (unità)}}{\text{Costo lavoro}} - \text{Costo orario} \cdot \text{labor content}}$$

Inoltre: • nelle linee di produzione non posso fare un task prima dell'altro

• " " " assemblaggio (Assembly line balancing) i vincoli sono minori e spesso è possibile

Ogni attività è composta da un insieme di tasks non ulteriormente frazionabili e, come detto, quasi sempre non invertibili

sequenza NON modificabile, posso solamente assegnare un task ad un lavoratore piuttosto che ad un altro

↳ in modo da rendere i process time (più corretto dire i cycle time) delle singole attività il più possibile uguali tra loro x minimizzare le attese (sbilanciamento linea) e quindi i costi

N.B. • Se il processo è demand-constrained il bilanciamento ha come unico effetto l'equità tra i lavoratori/macchine (ma il flow rate NON varia xK resta pari al tasso di domanda) ↳ spesso utile x questioni di manutenzione (fermo linea meno volte x ")

• Se invece il processo è Capacity-constrained il flow rate aumenta (in quanto è pari alla capacità del processo) e quindi diminuiscono i costi

(OSS. Esistono algoritmi ed euristiche a supporto di bilanciamenti complessi)

Se la domanda aumenta notevolmente devo scegliere se:

- uscire dal mercato
- appoggiarmi ad un terzo (va bene nel BT, nel caso di un picco, non di un aumento strutturale)
- aumentare la capacità (in questo caso la forza lavoro)

• 3 approcci x aumentare capacità processo:

① Replicazione linea: aggiungere altre linee in parallelo uguali a quella bilanciata

② Aumento Risorse x attività: assegno dei lavoratori aggiuntivi ad ogni attività

③ Variazione nella suddivisione task: modifico la ripartizione del lavoro inserendo altre risorse in serie in modo da diminuire i compiti (task) assegnati ad ogni lavoratore aumentandone la specializzazione (e quindi diminuendo i process times)

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 23 di 229

Cap. 10: Batching and other Flow interruption: Setup Times and the Economic Order Quantity Model

Nei processi ideali ogni X unità di tempo (process cycle time) una flow-unit entra nel processo e un'altra esce

Nella realtà il flusso di processo non è così costante e "smooth" per diversi motivi; i più importanti dei quali sono:

- i setups
- la variabilità nei tempi di processo e nei livelli di qualità

Setup

- Tempi di Setup (o changeover): tempi necessari quando cambio tipo di produzione
  - tempi di intervento sulle macchine (riattrezzaggio macchine)
  - tempi necessari a caricare e scaricare pezzi dalle macchine

- Non dipendono dall'attività che precede e da quella che segue
- Perturbano il sistema → quindi setup frequenti portano ad una diminuzione di capacità

Durante i setup la produzione è ferma (no pezzi prodotti)

Nel calcolare il cycle time, il tempo di setup va suddiviso (spalmato) tra tutti i pezzi di quella determinata tipologia (batch) (è sbagliato attribuirlo tutto al 1° pezzo di quella determinata tipologia)

quindi è importante decidere, quando ci sono tempi di setup, quanti pezzi produrre xk:

- tempi di setup hanno influenza sui tempi di processo
- all'aumentare delle parti (pezzi) che produco aumenta la capacità

In presenza di setup è quindi conveniente produrre a BATCH (lotti)

OSS. tutti i batch contengono lo stesso numero e tipo di flow units, le quali però possono essere dei set di componenti (ad es. 2A e 1B)

Interazione tra il batching e l'inventory:

All'aumentare della dimensione del batch, oltre ad aumentare la capacità, aumenta anche il WIP

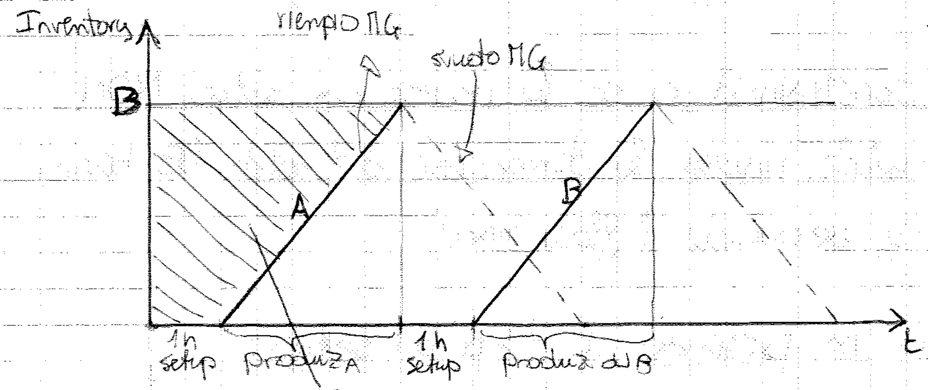
⇓

un aumento del WIP porta ad un aumento del Flow Time

⇓

Quindi produrre a batch relativamente grandi porta ad un mismatch tra il tasso di offerta e il tasso di domanda (risposta non veloce agli ordini dei clienti)

↳ per evitare, o meglio limitare, ciò occorre avere inventory



pezzi che stanno in magazzino (WIP) → in attesa di esser assemblati

Inventory medio:

$\bar{I} = \frac{B}{2}$ se $I_{iniziale} = 0$

$\bar{I} = \frac{B - B'}{2}$ se $I_{iniziale} = B'$

B grande:

- WIP alto
- alto costo WIP
- cycle time alto
- minimo spreco di capacità

B piccolo:

- WIP basso
- basso costo WIP
- cycle time basso
- grande spreco di capacità

⇒ Trade OFF tempi e costi (inventory) vs. capacità

OSS: Toyota Production Systems → "Mixed-model" ("heijunka") → "production smoothing"

⇓

produrre a batch di dimensione unitaria × sincronizzare produzione e domanda ed eliminare cycle inventory

Nota: in caso di mismatch tra, ad es, produzione ed assemblaggio è necessario inserire dell'idle time, durante il quale fermo la produz. di un componente × evitare un eccessivo accumulo di MG
↳ idle time usato × risincronizzare produzioni
↳ sfruttato anche × fare manutenzione

OSS. Il precedente modus operandi va bene quando c'è una sola risorsa con setup.
Nel caso in cui sia nota la domanda considero il processo demand-constrained; altrimenti il sistema è indeterminato e posso al massimo trovare le relazioni fra i vari batch

↓

Se ho più attività con setup impongo la loro capacità uguale alla domanda per calcolare B

• Impatto della varietà di prodotti su un processo con tempi di setup:

Aumentando la varietà nel mix di produzione (ossia iniziando a produrre nuove tipologie di prodotti) aumenteranno i setup per ogni production cycle

↓

questo porta ad una diminuzione della capacità

↓

x riportare la capacità al flow rate desiderato occorre aumentare la dimensione dei batch

↓

così facendo aumenterà il livello di inventory
↳ maggiori costi di MG

⟹ incompatibilità tra tempi di setup e varietà prodotti

↓

possibili soluzioni:

① offrire una varietà limitata di prodotti

② cercare di eliminare/minimizzare tempi di setup

↓

tempo di setup va ammortizzato sulle singole flow unit che compongono il batch
↳ quindi più riduco tempi di setup più potrò diminuire B (xk ho un tempo più piccolo da "spalmare")

↓

• Riduzione tempi di setup:

stampi, matrici

↓

SMED (Single Minute Exchange of Die): insieme di principi inventati in Toyota x ridurre tempi di setup
↳ l'obiettivo di tale metodo è di ridurre i tempi di setup ad un valore a singola cifra (ossia < 10 minuti) senza fare grossi investimenti in nuove tecnologie

14

Economie di scala legate ai costi legati ai setups

Si possono ridurre i costi unitari aumentando la dimensione del batch

batch grandi → diminuiscono i costi di setup
→ aumentano i costi di magazzino

⇒ Trade off : costi setup vs costi MG *

Modello del LOTTO ECONOMICO (Economic Order Quantity)

↳ modello ROBUSTO della gestione delle scorte

Si basa sull'ipotesi capacità ∞, ossia sovrabbondante (non vincolante)

↳ non significa che i tempi sono nulli

il problema della capacità non è un problema di tempi

posso considerare ogni prodotto indipendente dagli altri, in quanto avendo COSTI e non tempi non mi interessa più il MIX produttivo

Assunzioni

① Capacità ∞ → prodotti singoli

② Tempi (di approvvigionamento e produzione) → Ø

③ Tasso di domanda : COSTANTE (non varia)
DETERMINISTICA (ne conosco il valore)
CONTINUA (costante sul min intervallo di tempo immaginabile)
↳ 1000 pz/anno → 500 pz/semestre → 83.3 pz/mese ...

④ Costo d'acquisto indipendente dalla qtà acquistata (no sconti-quantità)

⑤ Costo di riordino (setup) " " " " (no economie di scala)

* Obiettivo : minimizzare i costi di MG e di riordino (setup) sotto il vincolo che l'inventory sia sempre > 0

• Dimensionamento del lotto (batch):

1. Individuare tutti i costi
2. " solo " " RILEVANTI x il nostro problema
3. Costruire una funzione di costo totale
4. Minimizzare tale funzione di costo

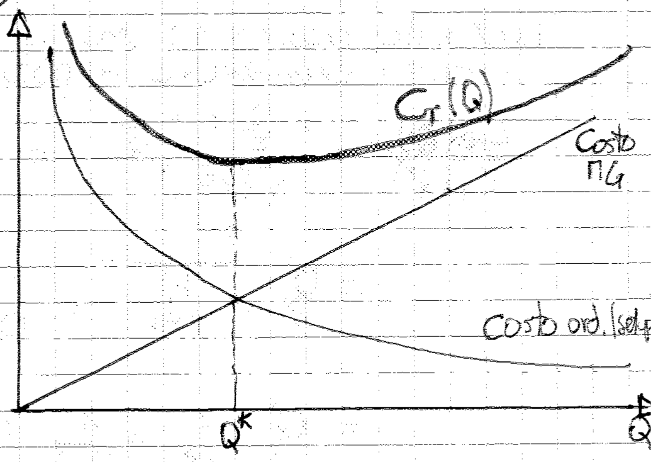
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 31 di 229

Sommando il costo (medio) di ordinazione nell'unità di tempo e il costo (medio) di magazzino nell'unità di tempo otteniamo:

• Costo totale per unità di tempo: $\boxed{C_T(Q) = h\cdot\frac{Q}{2} + K\cdot\frac{R}{Q}}$

OSS. il costo tot. d'acquisto non è incluso (xk NON rilevante x trovare Q^*)

C'è una quantità Q che minimizza il costo totale $C_T(Q)$



• Lotto economico EOQ (o Q^*):

quantità alla quale i costi di inventory e di setup si eguagliano

algebricamente: $\frac{\delta C_T(Q)}{\delta Q} = \frac{h}{2} - \frac{KR}{Q^2} = 0$

$$Q^2 = \frac{2KR}{h} \Rightarrow \boxed{Q^* = EOQ = \sqrt{\frac{2KR}{h}}} \rightarrow \boxed{T^* = \sqrt{\frac{2K}{Rh}}}$$ → frequenza di riordino ottimale

OSS. • se aumenta R → aumenta Q (maggior domanda da soddisfare)
• se aumenta h → diminuisce Q (in modo da minimizzare costi MG)
• se aumenta K → aumenta Q (in modo da ordinare meno frequentemente)

⇒ Sostituendo Q^* nella formula del costo totale:

$$C_T(Q^*) = \frac{h}{2}\sqrt{\frac{2KR}{h}} + \frac{KR}{\sqrt{\frac{2KR}{h}}} \Rightarrow \boxed{C_T(Q^*) = \sqrt{2KRh}}$$

↳ COSTO TOTALE MINIMO (ottimo)

Considerando anche il costo totale d'acquisto:

$$\boxed{\overline{C_T}(Q^*) = \sqrt{2KRh} + C\cdot R}$$

Ovviamente Q^* minimizzerà il $\boxed{\text{costo medio unitario} = \frac{C(Q^*)}{R} = \sqrt{\frac{2Kh}{R}}}$

N.B.: Costo unitario decresce all'aumentare del Flow Rate (domanda R)

⇒ c'è economia di scala nel processo di ordinazione

↳ operazione diventa più efficiente se la domanda cresce

La capacità di uno step sarà quindi:

$$Cap(B) = \frac{B}{Setup + B \cdot proc.time} = \frac{\text{qtà processata tra 2 stops}}{\text{durata stop} + \text{tempo per produrre 1 unità} \cdot \text{qtà processata tra 2 stops}}$$

⇒ essendo un processo continuo i vari step lavorano al passo del CdB

↓

Inserendo dei buffer si può quindi incrementare la capacità del processo rendendo più "smooth" il processo (meno interruzioni)

↳ in alcuni processi ciò non è possibile (es produz succo d'arancia, se inserisco buffer processo perde in qualità visto che vitamina C si ossida facilmente)

Cap 4 (Introduction to Distribution Logistics): Inventory Management with Deterministic Demand

Gestire la logistica di distribuzione richiede la coordinazione dei flussi di informazioni e materiali nella catena produttiva (supply chain) per acquisire efficienza (minimizzare i costi), ed efficacia (incontrare la domanda)

↓

La gestione delle scorte è un tema ampio e complesso, pertanto è utile, come prima cosa, identificare le variabili usate per classificare i vari problemi di inventory management e le loro soluzioni:

- Natura delle scorte e della supply chain (catena di generazione dell'offerta):

↓

- Le supply chains possono avere Lead Times deterministici o stocastici
  - ↳ nel secondo caso non c'è una stretta relazione tra acquisto/produzione e consegna → questo rende la pianificazione molto più complessa
- Supply chains possono essere:
  - single-item: interazioni deboli o nulle tra i prodotti → problemi di inventory indipendenti tra loro
  - multi item: quando ci sono interazioni tra i prodotti (es. economie di scala)
    - ↳ gestione più complessa xk capacità produttive, di trasporto ecc, non sono infinite come nel caso del single item
- problemi di pianificazione possono essere dinamici (se le decisioni vengono prese più volte nel periodo di tempo considerato) o statici (" prese una sola volta per tutte)

↓

questa distinzione dipende dal rapporto tra ciclo di vita del prodotto e Lead Times legati all'acquisto (LT di produzione)

17

Il caso più semplice è quello di un sistema deterministico con

- domanda stazionaria, deterministica e continua
- multi periodo (con ciclo di vita del prodotto infinito)
- Lead Times e livelli di inventory deterministici

⇓

Lotto Economico: sistema robusto di gestione delle scorte

↓

essendo robusto il problema non è se ci scostiamo dall'ottimalità, ma se NON sappiamo DI QUANTO ci scostiamo dall'ottimalità

↓

non conoscendo la domanda non posso conoscere i veri costi da minimizzare

↓

domanda ≠ vendite, per cui devo ricavare le info sui costi dai vari registri dell'azienda

La incertezza dovuta al non saper aggregare i dati nei registri e non alla volatilità/indeterminatezza

↓

non c'è variabilità ma incertezza

Rilassiamo ora le seguenti assunzioni irrealistiche:

1. - Lead Times $= \emptyset$
2. - velocità di riempimento MG infinita (Q consegnata interamente ad inizio T)
3. - Singolo prodotto
4. - costi lineari (NO sconti qtà)
5. - domanda stazionaria

1. Lead Times (tempi di consegna/trasporto) deterministici ma > 0 (MODELLO (Q,R))

↓

Se $LT \neq 0$, Q mi dà info su OGNI QUANTO emettere l'ordine ma NON su QUANDO emetterlo

↓

la domanda continua anche quando il MG è vuoto, pertanto non possiamo attendere che le scorte finiscano per ordinare, in quanto la qtà Q non è istantaneamente disponibile

↓

dovremo ordinare LT unità di tempo prima che le scorte si esauriscano, ossia non appena il livello di inventory raggiunge il cosiddetto:

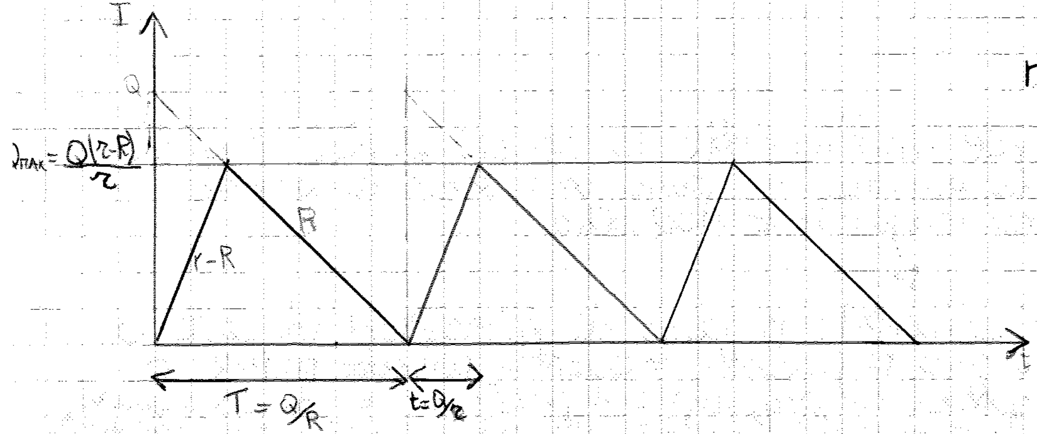
$$\boxed{\text{livello di riordino } \tilde{R} \leq R \cdot LT}$$

→ ossia riordiniamo quando abbiamo esattamente la qtà di scorte x soddisfare la domanda durante il LT

② Quantità Q consegnate progressivamente (velocità di riempimento del MG finita)

Si ha accumulo graduale del MG, ad es quando esso è servito direttamente dall'impianto di produzione

Questo cambia le dinamiche delle scorte nel MG, fa quindi variare anche la funzione costo e la quantità ottima Q^*:



r = capacità processo produzione = = tasso di approvvigionam = = n° di unità consegnate per unità di tempo

Quindi x ogni unità di tempo di t produco r e consumo R, pertanto livello MG crescerà al tasso (r-R)

Se $r \leq R$ il tasso (velocità) di produzione non è sufficiente (o è solo appena sufficiente nel caso r=R) per incontrare la domanda

Per il completamento del lotto Q sono necessari: $\frac{Q}{r}$ periodi e al completam. della produzione di un lotto Q il livello di inventory sarà massimo $Q_{MAX} = \frac{(r-R)Q}{r}$ ossia il lotto di produz. Q − la domanda occorsa durante il completamento del lotto $\left(Q - R \cdot \frac{Q}{r}\right)$.

Il Livello di Mg medio sarà quindi $\bar{I} = \frac{(r-R)Q}{2r}$

$$\boxed{C_{tot}(Q) = K \cdot \frac{R}{Q} + h \cdot \frac{(r-R)Q}{2r}}$$

chiamando $h' = h\left(\frac{r-R}{r}\right)$ ricaviamo Q^*:

$$Q^* = \sqrt{\frac{2KR}{h'}} = \sqrt{\frac{2KR}{h}\left(\frac{1}{\frac{r-R}{r}}\right)} \Rightarrow \boxed{Q^* = \sqrt{\frac{2KR}{h} \cdot \frac{r}{r-R}}}$$

detta Economic Production (o Manufacturing) Quantity:

$$EPQ = EMQ = EOQ \cdot \sqrt{\frac{r}{r-R}}$$

- Se $r = R$ → $Q^* = \infty$, ossia dovrei produrre continuamente (non a lotti) per soddisfare la domanda

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 39 di 229

b) "interazione vera" (vincolo sulla capacità degli ordini)

Il costo di riordino spesso dipende dal grado di utilizzo delle risorse amministrative e del personale del magazzino

ci sono praticamente sempre dei limiti di gestione degli ordini

Assumiamo che ogni prodotto sia ordinato da un fornitore diverso e che sia consegnato separatamente (no economie di scala).
Detto F il limite di ordini che possiamo gestire nell'unità di tempo:

i costi di riordino non sono più rilevanti per la nostra analisi e quindi l'unico obiettivo sarà quello di minimizzare i costi di MG sotto il vincolo del numero massimo di ordini gestibili

↳ otteniamo quindi un problema di ottimizzazione NON lineare (xk variabile Q è di grado -1)

$$\min \frac{1}{2}\sum_{i=1}^{N} h_i Q_i$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \frac{R_i}{Q_i} \leq F$$

supponiamo la soluzione INTERNA, ossia che non sta sulla frontiera (zero NON incluso) ← $Q_i > 0 \quad \forall i$

→ essendo un'ottimizzazione vincolata non basta fare semplicem. la derivata

essendo NON lineare non possiamo usare il metodo del simplesso

Per eliminare i vincoli NON lineari passiamo alla forma duale introducendo il

moltiplicatore lagrangiano λ = prezzo ombra = svolge il ruolo di una sorta di costo di riordino = costo marginale dell'aumento del numero di ordini o della diminuzione della Q

$$\boxed{\min \sum_{i=1}^{N}\left(\frac{h_i}{2}\cdot Q_i\right) + \lambda\left(\sum_{i=1}^{N}\frac{R_i}{Q_i} - F\right)}$$

- se λ troppo piccolo: ordiniamo piccoli Q troppo frequentemente, quindi n° ordini > F
- se λ troppo grande: capacità non pienamente utilizzata e livelli di inventario troppo alti

per trovare l'ottimo (minimo) bisognerà quindi fare $i+1$ derivate (una per ogni Q + la derivata risp. a λ):

$$\frac{\partial \mathcal{L}(Q_1, Q_2, \dots, Q_N, \lambda)}{\partial Q_i} = \frac{1}{2}h_i - \lambda\frac{R_i}{Q_i^2} = 0, \ \forall i \implies \boxed{Q_i^* = \sqrt{\frac{2\lambda^* R_i}{h_i}} \quad \forall i}$$

$$\frac{\partial \mathcal{L}(Q_1, Q_2, \dots, Q_N, \lambda)}{\partial \lambda} = \sum_{i=1}^{N}\frac{R_i}{Q_i} - F = 0 \implies \boxed{\sum_{i=1}^{N}\frac{R_i}{Q_i^*} = F}$$

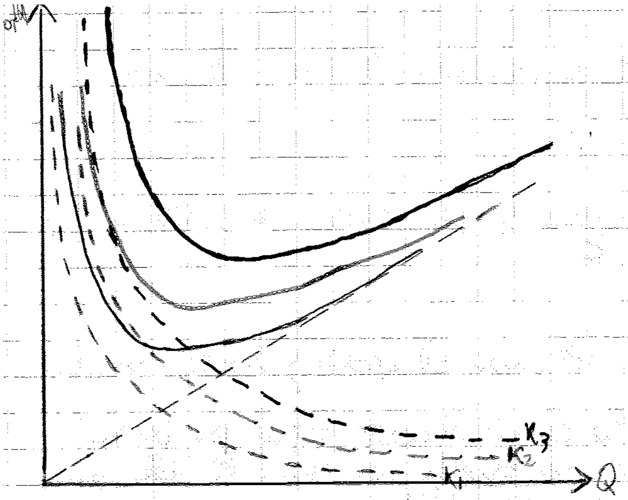
CONDIZIONI DI OTTIMALITÀ

da cui:

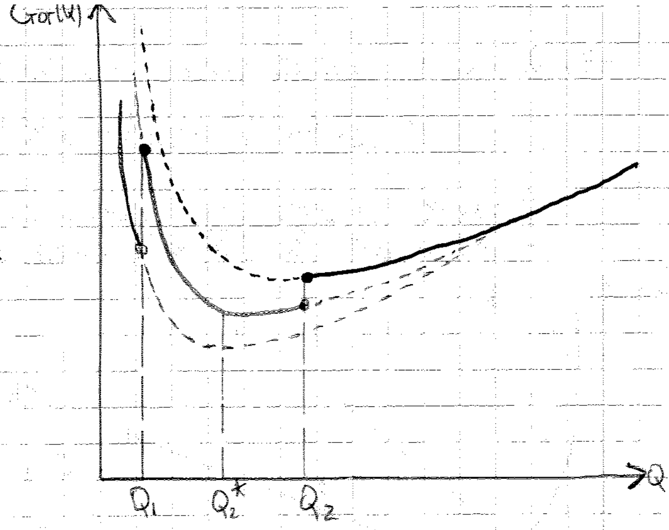
$$\lambda^* = \frac{1}{F^2}\left(\sum_{i=1}^{N}\sqrt{\frac{R_i h_i}{2}}\right)^2$$

quindi soluzione ottima sarà quella di fare più ordini possibile in modo da minimizzare Q^* e quindi anche i costi di MG

Le condizioni di ottimalità sono un sistema di equazioni non lineari che generalmente può essere risolto solo numericamente, tranne i casi in cui è facile trovare una soluzione in forma chiusa



$\Rightarrow$



Si nota come all'aumentare di K la funzione C_{TOT} subisce uno shift verso l'alto con un conseguente incremento della quantità ottima

$$opt(K_1) \Rightarrow Q_1 - 1$$
$$\Rightarrow opt(K_2) \Rightarrow Q_2^*$$
$$opt(K_3) \Rightarrow Q_2$$

Oltre a quello grafico vi sono altri 2 metodi per la ricerca dell'ottimo:

1) • calcolare lotti economici (Q^*) di tutte le funzioni
   • se rispettano vincolo sulla capacità → ammissibile → calcolo C_{TOT}
     se NON " " " " → NON " (Q^* fuori dall'intervallo di ammissibilità → mi sposto in uno dei 2 estremi di tale intervallo e quello sarà la mia Q con cui calcolare C_{TOT}
   • scelgo soluzione con C_{TOT} minore

2) metodo iterativo più breve:

   • parto dalla curva di costo tot. più bassa
   • ne calcolo l'EOQ.
   • se tale EOQ è ammissibile $\Rightarrow$ STOP $\Rightarrow$ questo è l'EOQ ottimo
     altrimenti passo alla curva di costo tot successiva (più in alto della prima) EOQ_2 e ripercorro gli step 2 e 3
   • scelgo EOQ minore tra quelli trovati prima dello STOP

Una procedura simile può essere utilizzata nel caso degli:

$\Rightarrow$ **Sconti Quantità** → in questi casi anche il costo d'acquisto unitario c diventa rilevante per la funzione di costo totale

$$\Rightarrow \boxed{C_{TOT} = K\frac{R}{Q} + \frac{1}{2}hQ + cR}$$

(dove c non sarà più fisso, bensì $c = c(Q)$)

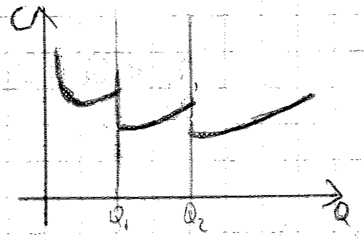
$\to h = h\% \cdot c(Q)$

funzione di C_{TOT} non varia più linearmente con Q, difatti h dipende da c e c dipende da Q (grandi batch potrebbero portare a riduzione costo d'acquisto)

21

Possiamo distinguere 2 tipologie di sconti:

- **All unit discount**: lo sconto si applica sull'intera quantità acquistata (es. 20% di sconto se il cliente acquista più di 100 unità)



La funzione costo presenta dei punti di discontinuità (salti) in corrispondenza dei quali il costo diminuisce

al raggiungimento della qtà minima richiesta per un dato sconto (Q_i), il costo totale d'acquisto (CR) e, di conseguenza, i costi di MG si riducono;

$$C = \begin{cases} C_0 & \text{se } 0 \le Q < Q_1 \\ C_1 & \text{"} \ Q_1 \le Q < Q_2 \\ \dots \\ C_n & \text{"} \ Q_n > Q \end{cases}$$

(con $C_0 \le C_1 < \dots \le C_m$)

inoltre parlando, l'ultima unità necessaria per ottenere lo sconto può avere un costo marginale negativo (acquistare Q_1 unità costa meno che acquistarne $Q_1 - 1$)

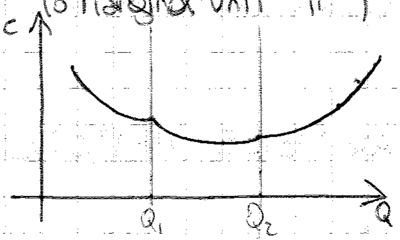
Per ogni intervallo di quantità si ha una differente funzione di costo totale, sulla base della quale possiamo determinare EOQ.

Per determinare la quantità ottima da ordinare (EOQ):

1. Calcolare $Q_i^* = \sqrt{\frac{2KR}{h_i}}$ $\forall$ intervallo i
2. Determinare il maggior Q_i^* AMMISSIBILE per il suo intervallo (ovvero che rispetti la condizione $Q_{i-1} < Q_i^* \le Q_i$) e calcolare il costo totale ad esso associato $C_T(Q_i^*) = \sqrt{2Kh_iR} + c_iR$
3. Determinare tutti i breakpoint (punti di frontiera) Q_J tali che $Q_J > Q_i^*$ e calcolare per ognuno di essi il costo totale $C_T(Q_J) = K\frac{R}{Q_J} + \frac{1}{2}h_iQ_J + c_iR$
4. Confrontare tutti i C_T calcolati agli step 2 e 3 e scegliere la quantità Q corrispondente al costo C_T minimo

OSS. Nel caso di All-unit discount, la quantità ottima è sempre un valore di EOQ di uno degli intervalli o un valore dei breakpoint

- **Incremental Discount**: lo sconto si applica solamente sulla quantità marginale (o Marginal Unit ") (es. 20% di sconto sulle unità oltre l'unità 100, del 30% dalla 200esima unità in poi...)



La funzione di costo tot. è continua ma presenta dei punti angolosi in corrispondenza delle variazioni di prezzo c_i (No salti/discontinuità)

Poiché varia il costo unitario c, varierà, per $\forall i$, anche il costo unitario di MG h

Per determinare il costo unitario di MG dovremo quindi utilizzare un costo MEDIO d'acquisto:

$$\bar{c} = \frac{C(Q)}{Q} = \frac{c_1Q_1 + c_2(Q_2 - Q_1) + \dots + c_n(Q - Q_{n-1})}{Q}$$

Rilassando l'assunzione di continuità non avrò più incognite (Q_i, C_i ..), bensì funzioni a tempo discreto ($Q_i(t)$, $C_i(t)$...)

a seconda del problema dovremo quindi scegliere come discretizzare l'orizzonte temporale e per forza ciò ci serviremo del:

TIME BUCKET (t): intervallo temporale all'interno del quale consideriamo la domanda costante (non ci interessa ciò che succede al suo interno: se scegliamo t = settimana non ci interessa conoscere ad es. la domanda giornaliera)

Pertanto non avremo più un unico parametro relativo alla domanda (R) ma ne avremo uno diverso per ognuno dei time buckets (d_t)

Di conseguenza serviranno quantità diverse in time buckets (t) diversi

Dovremo quindi determinare non più un'unica incognita (Q) bensì tante incognite quanti sono i time buckets:

X_t = qtà da produrre/ordinare nel time bucket t

LOT SIZING: problema di programmare la dimensione dei lotti produttivi in un orizzonte temporale discreto e in un ambiente produttivo manifatturiero, che nasce dalla necessità di contemperare nel modo migliore due voci di costo contrastanti (costi fissi (setup) e costi di immagazzinamento)

(trade off) infatti i costi fissi spingono verso pochi lotti di produzione grandi (in modo da non pagare troppo spesso tali costi), mentre per i costi di MG la situazione ideale sarebbe quella in cui si produce di volta in volta solo ciò che viene assorbito dal mercato, con molti piccoli lotti

Questo problema può essere risolto:

- con semplici regole che trattano i vari item in modo indipendente (Regole di lot sizing)
- oppure formulando un modello di programmazione lineare misto-intero (MILP)

obiettivo: soddisfare la domanda al minimo costo

Il costo totale d'acquisto (c·R) non impatta nella minimizzazione (a meno che non vi siano sconti quantità o che il costo unitario d'acquisto vari nei diversi time buckets)

Pertanto l'obiettivo è quello di minimizzare la somma dei costi di MG e costi di riordino in ogni intervallo t (o meglio di trovare la quantità X_t che minimizzi tale somma in t)

Pertanto riscriveremo il vincolo di Big-M nel modo seguente:

$$X_t \leq \left(\sum_{\tau=t}^{T} d_\tau\right) \cdot \delta_t$$

→ difatti poniamo $M_t = \sum_{\tau=t}^{T} d_\tau$ xk sappiamo che non dovremo comprare più prodotti di quelli necessari a soddisfare le domanda corrente e future (quindi la domanda dei time bucket $t, t+1, \dots, T$)

Ricorda:
- variabili presenti nella f.o. devono comparire nei vincoli o essere legate a quelle nei vincoli (se così non fosse significherebbe che non sono leve decisionali e verrebbero quindi poste pari a zero o ad ∞ in base alla f.o. (min/max))
- vincoli devono essere lineari (grado 1)
- f.o. deve " lineare (" ")

} NON moltiplicare variabili!

- se nel vincolo c'è $\sum_i$ non bisogna mettere $\forall i$

OSS. I modelli hanno soluzioni numeriche; non avremo quindi più una soluzione in forma chiusa

- UNCAPACITATED LOT SIZING (MULTI-ITEM):

$$\min \sum_i \sum_t (h \cdot I_{it} + A \cdot \delta_{it})$$

$$\begin{aligned}
\text{s.t.} \quad & I_{it} = I_{i,t-1} + X_{it} - d_{it} && \forall_{i,t} \\
& X_{it} \leq \left(\sum_{\tau=t}^{T} d_{i\tau}\right) \delta_{it} && \forall_{i,t} \\
& X_{it} \geq 0 && \forall_{i,t} \\
& I_{it} \geq 0 && \forall_{i,t} \\
& \delta_{it} \in \{0,1\} && \forall_{i,t}
\end{aligned}$$

Avendo assunto la capacità infinita, non ho vincoli di interazione tra i singoli item. Per tale motivo posso risolvere il problema come somma di tanti problemi a single-item (minor complessità).

NB Problemi UNCAPACITATED presentano sempre una soluzione ammissibile a differenza degli altri problemi

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 49 di 229

Vincolo di capacità mi forza interagire i prodotti, pertanto non potremo trattare ogni prodotto separatamente, com'era invece possibile nell'UNCAPACITATED

NB. Soluzione ottima del CLSP NON gode della proprietà di Wagner-Whitin (o per lo meno non sempre)

- Il vincolo di capacità inoltre definisce la regione di inammissibilità
  - ↳ in questo modello quindi potrei non trovare una soluzione ammissibile (ad es. xk anche lavorando a pieno capacità non riesco mai a soddisfare domanda)

Quando rilassiamo il problema nel continuo, per risolverlo ad esempio con un Branch & Bound, il vincolo big-M diventa computazionalmente inefficiente.

Per tale motivo M deve ~~essere~~ reso più piccolo possibile con uno dei seguenti due metodi:

- $M_{it} = \sum_{\tau=t}^{T} d_{i\tau}$ ⟶ vincolo diventa: $X_{it} \leq \left(\sum_{\tau} d_{i\tau}\right) \delta_{it}$ riferimento alla domanda future

oppure

- $M_{itm} = \frac{R_{mt}}{r_{it}}$ ⟶ vincolo diventa: $r_{im}\, X_{it} \leq R_{mt} \cdot \delta_{it}$ riferimento alla capacità delle risorse

Però la riduzione di M non è sufficiente a risolvere il problema di lot sizing in modo efficiente. È un problema NP-hard e, se includiamo i tempi di setup, è difficile anche solo trovare una soluzione ammissibile (se invece i " " sono trascurabili, possiamo facilmente costruire una soluzione ammissibile dal rilassamento continuo del modello, arrotondando le variabili di setup frazionarie).

Pertanto per risolvere tali sistemi ho 3 modi:

- implementare i risolutori commerciali nel nostro codice (poco efficiente)
- usare un'euristica o altri escamotage
  - ↳ • risolvere modelli in modo aggregato, riformulandoli come se fossero altri modelli che possiamo risolvere, o usando il cammino su grafo
    - ↳ Formulazioni forti: se la soluzione del rilassato è vicina alla soluzione ottima del problema intero
  - • suddividere il problema in problemi più semplici riducendone la complessità
    - ↳ ma se c'è interazione tra i prodotti spesso ciò non è possibile a causa del vincolo di capacità
      - ↳ una soluzione possibile potrebbe essere quella di inserire il vincolo nella f.o. passando quindi al problema duale e ottenendo un'architettura che mi permetterebbe di risolvere il problema x 1 singolo prodotto per poi controllare il resto tramite i moltiplicatori di Lagrange

Riassumendo, ho 4 metodi risolutivi:
- riformulare
- euristica
- algoritmo
- dualizzare il vincolo che dà interazione

⇒ <u>CLSP CON BACKORDER</u> (o Backlog o ordini inevasi)

In certi casi si ha un soddisfacimento dilazionato (in ritardo) rispetto all'accadimento della domanda

↓

avere backorder significa avere ordini consegnati in ritardo a causa del fatto che, ad un certo istante, la domanda cumulata è superiore alla produzione cumulata

↓

pertanto il MG virtuale sarà negativo (devo soddisfare degli ordini ed il MG fisico è vuoto)

↓

Per modellizzare tale situazione verrebbe spontaneo eliminare il vincolo $I_{it} \geq 0$ ma sarebbe sbagliato in quanto un MG fisico non può essere negativo

↓

Il livello del MG può quindi essere rappresentato come la differenza tra il valore del MG "on hand" ed il valore dei backorder:

$$I_{it} = I_{it}^+ - I_{it}^- \begin{cases} > 0 & \text{Mg fisico (pago i costi a magazzino)} \\ < 0 & \text{backorder (" eventuali sconti/penali)} \end{cases}$$

Introduciamo quindi il **PARAMETRO**: b_i = costo unitario per ogni unità di backorder

$$\min \sum_{t=1}^{T}\sum_{i=1}^{N} \left(h_i I_{it}^+ + b_i I_{it}^- + A_i \cdot \delta_{it}\right)$$

$$\begin{aligned}
\text{s.t.}\quad & I_{it}^+ - I_{it}^- = I_{i,t-1}^+ - I_{i,t-1}^- + X_{it} - d_{it} && \forall i,\ t = 2,\dots,T \\
& X_{it} \leq \left(\sum_{\tau=t}^{T} d_{i\tau}\right)\cdot \delta_{it} && \forall i,t \\
& \sum_{i=1}^{N}\left(r_{im} X_{it} + r_{im}^2 \delta_{it}\right) \leq R_{mt} && \forall m,t \\
& I_{it}^+,\ I_{it}^-,\ X_{it} \geq 0 && \forall i,t \\
& \delta_{it} \in \{0,1\} && \forall i,t
\end{aligned}$$

<u>OSS.</u> Le due componenti del MG (I_{it}^+ e I_{it}^-) hanno lo stesso segno nella f.o. poiché generano entrambi un costo

↓

(NB) Per tale ragione la soluzione ottima godrà della proprietà:

$$\boxed{I_{it}^+ \cdot I_{it}^- = 0}$$

→ ossia all'ottimo una delle due componenti sarà sempre nulla

↳ <u>NON è un vincolo!</u>

Nel caso non si voglia rendere obbligatorio il soddisfacimento della domanda:

- aggiungo la variabile v_{it} = qtà di i vendute in t
- " vincolo $v_{it} \le d_{it}$
- sostituisco d_{it} con v_{it} nel 1° vincolo
- riscrivo f.o. : $\max \sum_i \sum_t p_i \cdot v_{it} - \Big[\sum_i \sum_t h_i I_{it} + \sum_t \sum_j A_j \delta_{jt} + \sum_i \sum_j \sum_t (Q_{ij} \gamma_{ijt} + C_{ij} X_{ijt}) +$
  $+ f_i (d_{it} - v_{it})\Big]$

- MODELLO CON N FORNITORI (con 2 mezzi di trasporto diversi in capacità)

  ↳ soddisfacimento completo della domanda

Nuovi PARAMETRI: $A_j^{(1)}$ = costo trasporto veicolo grande da fornitore j
$A_j^{(2)}$ = " " " piccolo " " "
V_i = volume di i
w_i = peso di i
$V^{(1)}$ = limite in volume (capacità in volume) veicolo grande
$V^{(2)}$ = " " " (" " ") " piccolo
$W^{(1)}$ = " " peso (" " peso) " grande
$W^{(2)}$ = " " " (" " ") " piccolo

$$\min \sum_{i=1}^{N} \sum_{t=1}^{T} h_i I_{it} + \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{j \in J_i} C_{ij} \cdot X_{ijt} + \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{j \in J_i} Q_{ij} \gamma_{ijt} + \sum_{j=1}^{J} \sum_{t=1}^{T} A_j^{(1)} \delta_{jt}^{(1)} +$$
$$+ \sum_{j=1}^{J} \sum_{t=1}^{T} A_j^{(2)} \cdot \delta_{jt}$$

s.t.
$$I_{it} = I_{i,t-1} + \sum_{j \in J_i} X_{it} - d_{it} \qquad \forall i,t$$
$$I_{iT} = H_i \qquad \forall i$$
$$X_{ijt} \le \left(\sum_{k=t}^{T} d_{ik} + H_i \right) \cdot \gamma_{ijt} \qquad \forall i, t, j \in J_i$$
$$X_{ijt} = X_{ijt}^{(1)} + X_{ijt}^{(2)} \qquad \forall t, i, j \in J_i$$
$$\gamma_{ijt} \le \delta_{jt}^{(1)} + \delta_{jt}^{(2)} \qquad \forall t, j, i \in I_j$$
$$\sum_{i \in I_j} w_i X_{ijt}^{(1)} \le W^{(1)} \cdot \delta_{jt}^{(1)} \qquad \forall t, j \in J_i$$
$$\sum_{i \in I_j} w_i X_{ijt}^{(2)} \le W^{(2)} \cdot \delta_{jt}^{(2)} \qquad "$$
$$\sum_{i \in I_j} v_i \cdot X_{ijt}^{(1)} \le V^{(1)} \cdot \delta_{jt}^{(1)} \qquad "$$
$$\sum_{i \in I_j} v_i \cdot X_{ijt}^{(2)} \le V^{(2)} \cdot \delta_{jt}^{(2)} \qquad "$$
$$I_{it}, X_{ijt}, X_{ijt}^{(1)}, X_{ijt}^{(2)} \ge 0 \qquad \forall i, j, t$$
$$\gamma_{ijt}, \delta_{jt}^{(1)}, \delta_{jt}^{(2)} \in \{0,1\} \qquad \forall i, j, t$$

PARAMETRO: p_i = n° di item i producibili in un time bucket

$$\min \sum_{t=1}^{T}\sum_{i=1}^{N} (h_i I_{it} + A\cdot\delta_{it})$$

→ $\delta_{it}=1$ quando faccio setup di i in t

s.t.

$I_{it} = I_{i,t-1} + p_i x_{it} - d_{it}$ $\quad \forall i,t \quad$ (1)

$\sum_{i=1}^{N} x_{it} \le 1$ $\quad \forall t \quad$ (2)

mettendo uguaglianza obbligo a produrre in ogni time bucket aumentando i costi di MG

$x_{it} - x_{i,t-1} \le \delta_{it}$ $\quad \forall i,t$

$I_{it}, \delta_{it} \ge 0$ $\quad \forall i,t$

$x_{it} \in \{0,1\}$ $\quad \forall i,t$

diminuirei drasticamente l'efficienza del modello

NON è necessario che δ sia booleana: agli effetti pratici, trovandosi in f.o., nei casi in cui non può essere nulla, verrà posta pari ad 1 (in quanto f.o. minimizza)

difatti:

$\delta_{it} \ge x_{it} - x_{i,t-1}$

x_{it}	$x_{i,t-1}$	
1	1	→ produco i sia in t che in $t-1$ → $\delta=0$
0	0	→ NON " i né " " né " " → $\delta=0$
0	1	→ produco i solo in $t-1$ ma non in t → $\delta=0$
1	0	→ in $t-1$ non producevo i mentre in t lo produco → $\delta=1$ → pago setup

↳ unico caso in cui pago setup

Vincolo di capacità dato da $p_i x_{it}$ del vincolo (1) e dal vincolo (2)

se produco i, ne produco esattamente p_i unità e per ogni t posso produrre un solo item i

- Introduciamo ora i tempi di setup nei due casi:
  - tempi setup < time bucket
  - tempi setup multipli del time bucket

⇒ Lot Scheduling con SOLO TEMPI di setup < time bucket

↳ è sufficiente sottrarre, nel vincolo di MG, la quantità di prodotto "persa" durante il setup (conoscendo il tempo di setup conosco esattamente la qtà r_i non prodotta durante tale periodo):

$$\min \sum_{t=1}^{T}\sum_{i=1}^{N} (h_i\cdot I_{it})$$

↳ r_i = n° item producibili nel periodo in cui invece devo fare setup

s.t.

$I_{it} = I_{i,t-1} + p_i\cdot x_{it} - r_i\delta_{it} - d_{it}$ $\quad \forall i,t$

$\sum_{i=1}^{N} x_{it} \le 1$ $\quad \forall t$

$x_{it} - x_{i,t-1} \le \delta_{it}$ $\quad \forall i,t$

$I_{it}, \delta_{it} \ge 0$, $x_{it} \in \{0,1\}$ $\quad \forall i,t$

28

Cap. 1: The MRP Crusade

Abbiamo visto che per risolvere il problema del dimensionamento lotti (Lot sizing), oltre alla tecnica dell'EOQ, possiamo servirci di svariati modelli di programmazione lineare misto-intera.

Nei problemi con capacità e setup, la complessità computazionale è particolarm. elevata anche solo per trovare una soluzione ammissibile

↳ sono programmi NP-hard (Non polinomiali) nei quali, all'aumentare della grandezza del problema aumenta esponenzialmente la difficoltà di implementazione e risoluzione

Ma persino nei modelli UNCAPACITATED, se il modello è troppo grande (troppi time buckets) il tempo di risoluzione è inaccettabile in tempi pratici

difatti il Lot sizing è solo uno degli steps di metodi più grandi che vengono usati per cercare di gestire l'azienda nel suo complesso; pertanto sono auspicabili dei problemi di dimensionamento risolvibili in tempi brevissimi

A tale scopo sono stati sviluppati degli algoritmi euristici molto semplici noti col nome di: Regole di Lot sizing

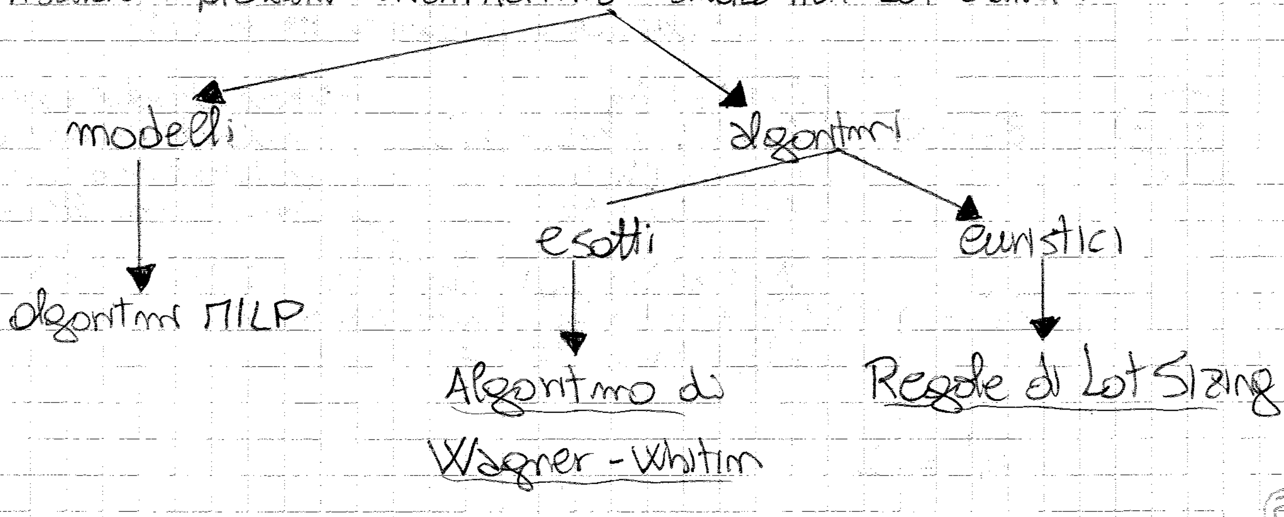
↳ procedimento EURISTICO: metodo di approccio alla soluzione dei problemi che non segue un percorso rigoroso ma, affidandosi all'intuito e allo stato temporaneo delle circostanze, consente di prevedere un risultato che resta da convalidare

Un algoritmo di approssimazione non dà una soluzione ottima ma dà un bound della regione di ottimalità, però è poco usato in quanto la sua complessità è paragonabile agli algoritmi esatti

↳ le euristiche invece consentono di trovare una soluzione molto facilmente (velocemente) ma non danno alcuna garanzia riguardo all'ottimalità

Le Regole di Lot sizing sono molto veloci ma molto poco flessibili (a differenza dei modelli)

Per risolvere i problemi UNCAPACITATED SINGLE-ITEM LOT SIZING:



$t=4$ da 1 a 4

$$Z_4^* = \begin{cases} A_1 + h_1(D_2+D_3+D_4) + h_2(D_3+D_4) + h_3D_4 \\ Z_1^* + A_2 + h_2(D_3+D_4) + h_3D_4 \\ Z_2^* + A_3 + h_3D_4 \\ Z_3^* + A_4 \end{cases}$$

supponiamo che sia il costo minimo

$J_4^* = 3$

$t=5$ da 1 a 5

$$Z_5^* = \begin{cases} A_1 + h_1(D_2+D_3+D_4+D_5) + h_2(D_3+D_4+D_5) + h_3(D_4+D_5) + h_4D_5 \\ Z_1^* + A_2 + h_2(D_3+D_4+D_5) + h_3(D_4+D_5) + h_4D_5 \\ Z_2^* + A_3 + h_3(D_4+D_5) + h_4D_5 \\ Z_3^* + A_4 + h_4D_5 \\ Z_4^* + A_5 \end{cases}$$

qst 2 alternative sono eliminabili dato che dal passo $t=4$ so già che produrrò in $t=3$

quindi nei prossimi step scarto le alternative che prevedono di immagazzinare la qtà D3 in MG (escludo anche le alternative che prevedono ordine prima di $t=3$)

in qst modo riduco decisamente il numero di alternative (quando calcolerò J^* potrò tagliare ulteriori alternative)

Con l'algoritmo di W-W si arriva sempre ad una soluzione (ottima), che viene ricavata nel seguente modo:

t	1	2	3	4	5	6	7	8	9	10
D_t	10	5	5	3	1	10	10	2	1	5
X_t	15		8		21			2	6	

$J_1^*=1$ $J_2^*=1$ $J_3^*=3$ $J_4^*=3$
$J_5^*=3$ $J_6^*=5$ $J_7^*=5$ $J_8^*=8$
$J_9^*=8$ $J_{10}^*=9$

Per trovare le X_t, ossia la soluzione ottima, parto dall'ultima J^* e scrivo in corrispondenza del periodo da esso indicato la somma delle domande di quel periodo e dei successivi (in questo caso essendo $J_{10}^*=9$ avrò $X_9 = D_9 + D_{10} = 5+1 = 6$). Fatto ciò passo al J^* subito precedente e ripeto quanto fatto per l'ultimo J^* (ossia $X_8 = D_8 = 2$)

Sebbene l'algoritmo di W-W permetta di avere un numero di esplorazioni che cresce linearmente con il numero di iterazioni, nel caso di problemi con un orizzonte temporale molto ampio esso sarà lento, oltre che complesso.

Per tale motivo si tende ad utilizzare le Regole di Lot Sizing a discapito della certezza dell'ottimalità della soluzione

Es. $T = 3$

t	1	2	3	4	5	6	7	8	9	10
D_t	10	5	5	3	1	10	10	2	1	5
X_t	20			14			13			5
I_t	10	5	–	11	10	–	3	1	–	–

indice semplicem. qnt resta in MG a fine periodo ← I_t

$C_{TOT} = 4 \cdot 100 + 2(10+5+11+10+3+1 = 480$ €

↳ serve a notare che si produce sempre quando MG è vuoto (propr. W-W)

(1b) $T_{EOQ\,mod}$: caso particolare della FOP in cui determino il valore ottimale di T sfruttando le formule dell'EOQ:

$$T_{EOQ\,mod} = \frac{Q}{\bar{D}} = \sqrt{\frac{2A}{K\bar{D}}}$$ dove $\bar{D}$ = domanda media

(2a) FIXED ORDER QUANTITY (FOQ) → duale della FOP

Prevede di produrre/ordinare sempre la stessa quantità ogni qual volta si produce/emette un ordine.

↳ si sceglie una Q fissa e si ordina/produce sempre Q o multipli di Q

↳ vantaggi :
- quando, all'interno dello shop, vengono usati trasportatori, carrelli o altre attrezzature x trasportare i jobs è utile che questi ultimi abbiano tutti dimensioni pari (o multiple) alle dimensioni di tali attrezzature
- fissare la dimensione dei jobs influenza il numero di setups

Es. $Q = 4$

t	1	2	3	4	5	6	7	8	9	10
D_t	10	5	5	3	1	10	10	2	1	5
X_t	12	4	4	4	–	12	8	4	–	4
I_t	2	1	–	1	–	2	–	2	1	–

→ tutti multipli di 4

OSS. La FOQ NON GODE della PROPRIETÀ di W-W

↳ qst significa che potrebbe portare ad un aumento dei costi di MG non dovuti a setup (inefficienza)

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 63 di 229

3b PART-PERIOD BALANCING (PPB)

È una regola che combina le assunzioni del paradigma di Wagner-Whitin con i meccanismi dell'EOQ

Infatti l'idea della PPB è quella di bilanciare i costi di MG e di setup (condizione di ottimalità del lotto economico)

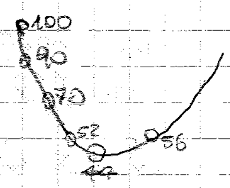
Part-period = numero di parti nel lotto (domanda) • n° di periodi in cui restano in MG

↳ costo di MG = part-period • h

Es. $A = 100$
$h = 2$

	t	costo setup	costo magazzino	$\lvert\Delta\rvert$ = setup-MG
			perché in t=1 produco	
lotto 1 ←	1	100	0	100
lotto 2 ←	2	100	5·2	90
	3	100	5·2+5·2+5·2	70
	4	100	30+3·2+3·2+3·2	52
	5	100	48+1·2+1·2+1·2+1·2	44 → punto di minimo!
	~~6~~	~~100~~	~~56+5(10·2)~~	~~56~~
	6	100	0	100
	7	100	10·2	80
	8	100	20+2(2·2)	72
	9	100	28+3(1·2)	66
	10	100	34+4(5·2)	26

da t=6 in poi mi allontano dall'obiettivo, per cui produco di nuovo in t=6 e riporto il conteggio



⇒ produco/ordino in t=1 per i periodi 1,2,3,4,5. In t=6 produrrò di nuovo per i periodi che vanno da t=6 a t = periodo in cui trovo un altro minimo locale

Quindi PPB consiste nella ricerca dell'ottimo (minimo) locale
↓
in qst punto costi di setup ≈ costi di MG
↓
alla fine del periodo corrispondente al punto di minimo locale avrò I = 0
⇓
OSS. Pertanto la PPB GODE delle PROPRIETÀ di W-W

3c LEAST UNIT COST (LUC) → ricerca del costo unitario minimo

Simile alla PPB ma, in questo caso, invece di bilanciare i costi di MG e setup, calcoliamo il rapporto tra C_{TOT} (da 1 a t) e quantità cumulata (sempre da 1 a t)

↳ In t = 1 produrremo/ordineremo una quantità tale da soddisfare il fabbisogno fino al periodo corrispondente al punto di minimo locale (rapporto minimo)

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 65 di 229

MATERIAL REQUIREMENTS PLANNING (MRP)

I modelli trattati finora hanno un grosso limite: non funzionano se applicati in azienda in quanto ci permettono di gestire SOLO LE SCORTE DI PRODOTTI FINITI, ossia dei prodotti di livello più alto.

Nelle situazioni reali quasi sempre tali prodotti sono il risultato dell'assemblaggio di più componenti, ciascuno dei quali deve essere prodotto o acquistato prima della fase di assemblaggio

pertanto ci troviamo quasi sempre ad avere a che fare con sistemi multilivello nei quali sono presenti MAGAZZINI INTERMEDI

tali magazzini NON sono gestibili con i modelli di lot sizing in quanto questi ultimi necessitano, come parametro in ingresso, della domanda ESOGENA (che viene dall'esterno) e non della domanda ENDOGENA (interna)

le scorte dei MG intermedi non sono indipendenti tra loro, bensì interagiscono a causa della limitata capacità di assemblaggio al termine del processo

Alla fine degli anni '60 Joseph Orlicky, sviluppatore dell'IBM, trovò un metodo per gestire i magazzini multilivello, affidando ai computer i compiti di schedulazione e controllo dell'inventory

Il boom dell'MRP si verificò negli anni '70 e da quel momento è il maggior componente della quasi totalità degli approcci computerizzati alla gestione della produzione, inclusi:

- Manufacturing Resources Planning (MRP II)
- Business Resources Planning (BRP)
- Enterprise Resources Planning (ERP)
- Supply Chain Management (SCM)

Prima dell'MRP, la maggior parte dei sistemi di controllo della produzione erano basati su diverse versioni dei punti di riordino statistici (EOQ, modelli...), ma Orlicky & Co. capirono che tale approccio era molto più adatto per i prodotti finiti che per i componenti. La motivazione, come accennato in precedenza, sta nel fatto che la domanda per i prodotti finiti si origina all'esterno del sistema ed è quindi soggetta ad incertezza. Ma dal momento che i componenti sono usati x produrre i prodotti finali, la domanda di tali componenti sarà funzione della domanda di prodotti finiti ed è quindi NOTA per ciascuna schedulazione dell'assemblaggio finale.

Trattando questi due tipi di domanda in modo equivalente, come si fa nei sistemi a punto statistico di riordino, si ignora la dipendenza di una domanda dall'altra e ciò porta ad inefficienze nella schedulazione della produzione (spesso fatali x l'azienda).

Si distingue quindi tra:
- domanda indipendente (esogena): quella x prodotti finali e componenti da usare come pezzi di ricambio
- domanda dipendente (endogena): " x componenti

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 67 di 229

① INPUTS:

a) Previsione della domanda dei prodotti finiti
b) Distinta base
c) Livello di MG corrente
d) Info necessarie a specificare le politiche di produzione: - item master File
- master production schedule
- inventory status File

• The Master Production Schedule (MPS)

L'MPS è la fonte di domanda per il sistema MRP.
↳ Fornisce la quantità e le due dates per tutte le parti che hanno domanda indipendente (prodotti finiti e componenti di ricambio)

MPS contiene un set di records contenenti (per ogni purchase orders)
- codice prodotto
- quantità necessaria
- due date

Queste informazioni sono usate dall'MRP per ottenere i Fabbisogno Lordo (Gross Requirements) che da inizio alla procedura MRP

L'MPS tipicamente usa il codice prodotto per collegare agli item le altre info incluse nell'item master file

• The Item Master File contiene:

- BILL OF MATERIAL (Distinta Base):
descrive la relazione tra i prodotti finiti e gli item di livello più basso

Il BOM contiene, per ogni end-item (prodotto finito), i componenti e le quantità che sono direttamente richiesti per produrre (assemblare) l'end-item

Per facilitare l'elaborazione dell'MRP, ad ogni item presente nel BOM è associato un LOW-LEVEL CODE (LLC). I prodotti finiti (ossia gli item di livello più alto) hanno LLC=0. I componenti usati solamente ~~direttamente~~ per assemblare il prodotto finito hanno LLC=1. " " " " " " i componenti con LLC=1 hanno LLC=2. Se un componente è usato nell'assemblaggio di un prodotto finito ma anche nell'assemblaggio di un componente di livello più basso (es LLC=2) avrà l'LLC relativo al più basso livello (ossia sarà LLC=3). Proprio per tale motivo si chiama LOW LEVEL CODE (in questo assume il valore relativo al livello più basso).

La maggior parte degli MRP commerciali (es SAP) includono un BOM processor che svolge le funzioni di:
- assegnare automaticamente gli LLC
- generare le liste di "goes-into" (dove le parti sono usate)

• **CHANGE NOTICES**: indicano le modifiche ai jobs esistenti, come i cambiamenti di due dates o di priorità

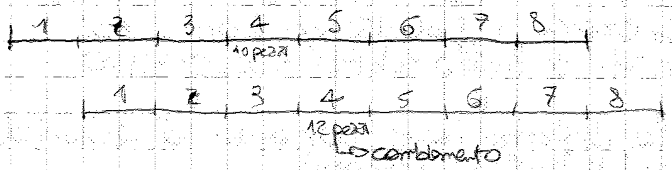
↓

CAMBIAMENTI

expediting: anticipare una due date
deferring: posticipare " " " (finale in ritardo)

• **EXCEPTION REPORTS**: usati x notificare discrepanze tra quanto previsto nel piano e quanto si manifesta

Assumendo che time bucket = 1 sett e che si voglia pianificare per 2 mesi:



L'operazione di pianificazione viene quindi ripetuta al termine di ogni time bucket in modo da apportare i cambiamenti necessari

Alla 2° settimana ripianifico le 7 settimane che avevo già pianificato correggendo eventuali errori, problemi o discrepanze.

Più si accorcia l'orizzonte temporale più la mia pianificazione sarà precisa, in quanto i dati forniti dall'MPS saranno più precisi

③ LOGICA DI FUNZIONAMENTO:

La procedura dell'MRP è basata sull'uso di tabelle, una x ogni codice; compilate partendo dal codice di livello più alto (quindi con LLC più basso).

La procedura base è piuttosto semplice e si può logicamente suddividere in 4 fasi o steps:

1. **NETTING**: si determina il Fabbisogno Netto sottraendo l'ON-HAND inventory e le consegne attese (SR) dal Fabbisogno Lordo.

   Il Fabbisogno Lordo per gli items con LLC = 0 è dato dall'MPS, mentre quello per gli items di livello inferiore sono il risultato delle iterazioni degli MRP precedenti
   ↳ ultima riga (OR) della tabella di un codice LLC = 0 diventa la prima riga (FL) della tabella del corrispondente codice (LLC = 1)

   Nel momento in cui l'OH inventory diventa < 0 si avrà Fabbisogno Netto

2. **LOT SIZING**: utilizzando uno dei metodi euristici visti (regole di lot sizing) o un modello, si divide la domanda netta (FN) in appropriati lotti x formare i jobs
   ↳ si passa da riga FN a riga OP

35

Esempio di tabelle:

t	1	2	3	4	5
FL/GR					
SR/CA					
OH (OH_0)					
FN(NR)					
OP(POR)					
OR(POR$_r$)					

1° riga: Fabbisogno Lordo (Gross Requirements): indica quanto ci serve in ogni t.b. del componente per cui stiamo facendo la tabella
↳ FL ≡ MPS se la tabella si riferisce al prodotto finito
↳ FL componenti NON corrisponde con la domanda esterna (es. se in 5 mi chiedono 10 sedie io i 10 schienali devo produrli prima del t.b. 5)

2° riga: Consegne Attese (Scheduled Receipts) consegne degli ordini effettuati nei periodi precedenti

3° riga: On-Hand Inventory (o Projected OH) proiezione di quello che sarà il livello di MG fisico al termine del time bucket

4° riga: Fabbisogno Netto (Net Requirements) = FL − SR − OH
↳ domanda interna
↳ $FN_t = \min\{FL_t, \max\{0, -OH_t\}\}$
↳ già depurato del FL_t

5° riga: Planned Order Receipts (Ricevimento Ordine Pianificato) indica la qtà da ordinare/produrre (ricavata dal Lot Sizing) in corrispondenza del time bucket in cui è necessaria (due dates)

6° riga: Planned Order Releases (Rilascio Ordine Pianificato) indica la qtà da ordinare/produrre in corrispondenza del t.b. in cui deve essere emesso l'ordine/rilasciata la produzione (OR = OP shiftato)

Come affermato in precedenza il sistema MRP genera 2 tipi di output:

• PIANO: dato dall'insieme delle righe OR (Planned Order Releases) → JOB POOL di tutte le tabelle generate dall'esplosione della distinta base
↳ ci dice QUANTO, QUANDO e COSA mandare in produzione nei vari time buckets

• CAMBIAMENTO: generalmente riferito alle Consegne Attese (SR)
infatti le procedure MRP prevede che, per soddisfare la domanda, come prima cosa si attinge dall'OH inventory; una volta esaurito l'OH si ricorre alle Scheduled Receipts e, solo infine, si procede con nuovi ordini.
Pertanto bisogna correggere i piani in modo tale che non vi siano:
– consegne attese nei time buckets successivi a quello del ricevimento del 1° ordine pianificato (dato che le CA, essendo figlie di un piano precedente, sono state rilasciate in precedenza)
↳ necessario cercare di anticipare CA → spesso ciò NON è possibile a causa della scarsa flessibilità dei fornitori,

la consegna da parte del fornitore non arriva in tempo)

↓

quindi SS usate per "recuperare un ritardo" del quale mi sono accorto troppo tardi

↓

tali SS richiedono una produzione intenzionale di quantità per le quali non vi è richiesta da parte dei clienti

↓

pertanto le SS fanno aumentare i costi di MG

↳ il livello desiderato di Scorte di Sicurezza si decide tenendo conto della variabilità della domanda e dell'incertezza sulla produzione

↓

Se si stabilisce che il livello di SS debba essere X, non appena ne si usa anche solo una parte per scongiurare un rischio di infeasibility, facendo quindi scendere tale livello al di sotto della soglia X, si deve immediatamente rilasciare un ordine per ripristinare tale livello.

↳ Ho più alternative per tener conto delle SS all'interno del sistema MRP, ad esempio posso:

- Non segnarle tra le quantità OH_0
- Segnare ad es $OH_0 = 50$ di cui $SS = 20$ ⇒ quindi OH disponibile $= 30$

- **Lead Time di Sicurezza (Safety Lead Time):**

può essere usato per proteggersi dall'incertezza sulla produzione (ossia incertezza sui tempi)

↓

Per tutelarmi da problemi che si possono verificare in produzione (come assenteismi, guasti, rotture, manutenzioni straordinarie...) posso partire in anticipo con la produzione (o emettere in anticipo gli ordini)

↓

Per implementare tale idea nel sistema MRP, se il piano prevede $LT = 2$, posso sostituirlo con $LTS = 4$ (in modo da rilasciare gli ordini 2 periodi prima rispetto a quanto pianificato in precedenza)

↓

Se non si verificano ritardi in produzione, per un certo numero di time buckets, pagheremo più costi di MG del dovuto

N.B. Un eccesso di SS e di LTS, poiché entrambi "mentono" al sistema, può portare ad un piano "percepito" infeasible nonostante esso sia nella realtà feasible. Tale problema porterà il sistema a rifiutare (respingere) alcuni ordini da parte dei clienti benché essi possano essere soddisfatti

In un sistema MRP vi sono molte cose che possono andare per il verso sbagliato: jobs possono finire in ritardo, alcune parti possono essere scartate, domanda può cambiare ecc...

I sistemi MRP hanno varie caratteristiche/strumenti che assistono colui che pianifica in situazioni in cui vi sono dei cambiamenti di condizioni, ad es. il:

- **Pegging**: che consente di tenere traccia dell'attribuzione di una parte (item specifico) al prodotto finale (ossia al cliente)

  fornisce un collegamento tra il fabbisogno lordo di un item e tutte le sue fonti di domanda

  Questo è molto utile quando un codice compare in più distinte base

  Se un determinato codice è richiesto per l'assemblaggio di più parti, tramite il pegging posso risalire alla parte in cui è stato effettivamente usato

  Uno degli usi del pegging è infatti nello:

- **Bottom-Up Replanning**: fase, anche chiamata "ripianificazione dal basso", che serve ad es. ad investigare sull'impatto che ho un ritardo di una consegna attesa sulle consegne al cliente (due dates).

  Se non riesco a soddisfare tutta la domanda xk ho parti insufficienti di un determinato codice, userò quel determinato codice x soddisfare la domanda (gli ordini) che più mi conviene

  tramite il pegging colui che pianifica può risalire alla fonte di domanda che risulta in un dato OR

Quelle sin qui trattate sono le peculiarità dell'MRP, ossia gli aspetti di cui tener conto per non rischiare il fallimento. Analizzeremo ora:

Problemi dell'MRP:

Sono problemi intrinsechi dell'MRP di cui dobbiamo essere consapevoli e che dovremo attenuare dato che non possono essere eliminati

Fondamentalmente l'MRP è un sistema estremamente costoso, NON robusto, difficile da gestire. Una delle difficoltà principali consiste nell'individuare i dati esatti da inserire.

I 3 problemi più gravi sono:
1. Capacity Infeasibility dei piani MRP
2. Planned Lead Times troppo lunghi
3. Nervosismo del sistema

Supponendo, ad es. che il tempo di produzione medio sia di 3 sett, quindi LT=3, con una deviazione standard di 1 sett ($\sigma = 1$), per mantenere un alto livello di servizio al cliente bisogna porre LT = 5 sett (2 volte la st. dev. x coprire ≈ 95% dei casi)

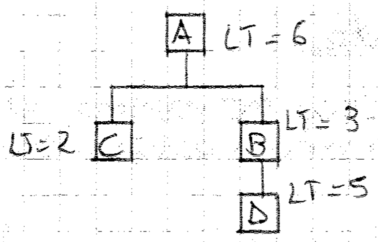
Cioè, per compensare la variabilità, si tende a scegliere delle stime pessimistiche dei LT

↳ Lead Time esasperato: comprende 2-3 volte la deviazione standard in modo da portare il livello di servizio al cliente (probabilità di rispettare le due dates) oltre il 99%

Porre LT = tempo di produzione medio, porterebbe il livello di servizio al cliente al di sotto del 50%.

Però, questa politica dei lead times esasperati, costringe il manager a dover pianificare per parecchie settimane.

Es.



Si nota che dalla produzione di D al rilascio di A passano 14 sett.

pertanto si dovrà pianificare x ALMENO 14 sett

il problema sta nel fatto che più avanti sposto l'orizzonte temporale (più pianifico a lungo termine) meno accurate saranno le previsioni a causa della maggiore incertezza

pertanto sarò costretto ad aggiungere delle SS e/o dei LTS che a loro volta sposteranno ancora più avanti l'orizzonte temporale della pianificazione

Si entra in un "loop" rischioso

Può causare dimensioni smisurate dei MG

difatti, per la legge di Little, all'aumentare dei LT aumenterà l'inventory e con esso i costi di MG

Soluzione: grazie ai calcolatori più evoluti non si è più vincolati a t.b. grandi e, dato che i LT sono sempre multipli dei t.b, il problema è stato attenuato, benché i LT rimangono ineliminabili nell'MRP

3 Nervosismo:

Si verifica nervosismo in un sistema MRP quando un cambiamento minimo nel MPS (quindi nella domanda o FL) causa grandi cambiamenti nel Rilascio degli Ordini Pianificati (OR).

in modo diverso. Ad esempio:

1° time fence {t = 1...4}: congelato (nessun cambiamento possibile)
2° " " (t = 5...7): restrizioni meno rigide
(ad es. sono ammessi cambiamenti sulle opzioni ma non sulle quantità)
3° " " (t = 8...12): minime restrizioni
(ad es. sono accettati cambiamenti non solo per le opzioni ma anche x le quantità se tutti i componenti sono disponibili nell'OH)

4) Firm Planned Orders (FPO):

a differenza del congelamento e dei time fences, i FPO fissano solo gli OR e non l'intero piano.
Fissando ad esempio i primi OR sposto in avanti gli ordini successivi e elimino il nervosismo del sistema almeno sul breve termine (dove è più dannoso).

Il difficile sta nel capire quali ordini bloccare

OSS. Ovviamente l'uso dei FPO e del time fencing porta i time buckets congelati del piano ad essere meno reattivi ai cambiamenti della domanda. Inoltre gli FPO sono particolarmente fastidiosi in quanto devono essere gestiti manualmente dal manager

Manufacturing Resources Planning (MRP II)

Col tempo, sono state sviluppate nuove procedure aggiuntive nell'ottica di eliminare, o quantomeno attenuare, i principali problemi dell'MRP

Queste procedure aggiuntive sono state poi incorporate in più ampi sistemi conosciuti come Manufacturing Resources Planning (MRP II)

oltre ad affrontare le carenze dell'MRP, l'MRP II presenta numerose altre funzioni che lo rendono un vero e proprio sistema per la gestione della produzione

Tra queste funzioni aggiuntive troviamo:
- demand management
- Forecasting
- Capacity planning
- Master Production Scheduling (MPS)
- Rough-Cut Capacity Planning (RCCP)
- Capacity Requirements Planning (CRP)
- Dispatching
- Input/output Control

Analizzeremo ora la gerarchia base di un MRP II:

• Aggregate Production Planning: usato x determinare i livelli di produzione, personale, inventory, straordinario ecc, sul lungo termine
↓
livello di dettaglio tipicamente mensile e molto aggregato
↓
vengono spesso usate tecniche di ottimizzazione come la programmazione lineare

Una volta terminata la pianificazione aggregata passo agli ordini (Firm orders) e alle previsioni di breve termine:

• Short-term Forecasting: converte le previsioni di lungo termine in aggregato (famiglie di prodotti), in previsioni di breve termine su singoli prodotti finiti

⇒ Pianificazione a Medio-Periodo:

A livello intermedio, troviamo la maggior parte delle funzioni di pianificazione della produzione:

• Demand Management: riceve in input le previsioni di breve e lungo termine e i firm orders. Ha la funzione di convertire le previsioni aggregate di lungo termine in una previsione dettagliata tracciando gli ordini dei singoli clienti
↓
Cerca quindi di capire quanto si dovrà immettere nel mercato nel medio periodo
↓
dà come output un insieme di ordini "presenti" dei clienti più una previsione di ordini previsti
↳ con l'avanzare del tempo gli ordini previsti vengono "consumati" dagli ordini presenti
↓
Ciò è portato a termine con la tecnica dell' Available to Promise (ATP)
↓
permette a colui che pianifica di sapere quali ordini dell' MPS sono già stati impegnati e quali invece è possibile "promettere" ai nuovi clienti
↓
partendo dalle previsioni, dagli ordini e dall' MPS, l'ATP (combinato con un MPS capacity-feasible) permette di promettere una data di consegna (due dates) realistica ai clienti (anche se cmq grezza, non molto precisa)
↓
(N.B.) • Posso conoscere la quantità ATP solo in corrispondenza dei time bucket in cui l'MPS mi dice che arriverà un ordine o in $t=1$ se conosco il MG-On-hand attuale
↳ ossia solo negli istanti in cui ho la certezza del numero di pezzi disponibili; nei periodi per i quali non conosco la

- **Master Production Scheduling (MPS):**

prende la previsione della domanda e i firm orders dal modulo demand management e, usando limiti di capacità aggregata, genera uno schedule di quelle che saranno le quantità richieste, nel medio periodo, di codici di livello più alto (non più aggregati)

↓

queste costituiscono le "domande" usate dal sistema MRP

↓

Pertanto l'MPS conterrà un "order quantity" in ogni time bucket per ogni item con domanda indipendente

↓

MPS, quindi, disaggrega l'output della pianificazione aggregata sfruttando le info fornite dal demand management, e fornisce il fabbisogno lordo di prodotti finiti nel medio periodo già mediato dalla capacità disponibile e dei piani precedenti

↓

disaggregando le previsioni sulle famiglie di prodotti in previsioni sui singoli prodotti finiti, devo però tener conto delle interazioni di questi ultimi da un punto di vista di interazioni e di capacità

↓

pertanto MPS dovrà essere seguito da un check sulla capacità (sulla base dell'MPS stesso) che, se darà esito negativo, obbligherà a correggere l'MPS.

↓

Come vedremo tra poco però, questo non mi garantirà di evitare situazioni di infeasibility

- **Rough-Cut Capacity Planning (RCCP):** usato per fare un check, dal punto di vista della capacità, di alcune risorse critiche in modo da garantire l'ammissibilità (feasibility) dell'MPS

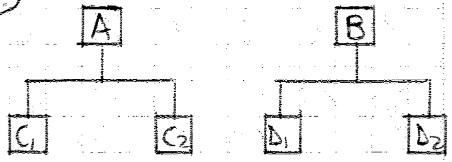
↓

RCCP fa uso della cosiddetta Bill of Resources

↓

fornisce il numero di ore necessarie ad ogni risorsa critica per produrre un particolare prodotto finito

Es.



t	1	2	3	4	5	6	7	8
A	10	15		20	5	10		5
B	25		5	5	10		10	20

Bill of Resources:

Work center	A	B
X100-K	3h	2h

→

t	1	2	3	4	5	6	7	8
A(h)	30	45		60	15	30		15
B(h)	50		10	10	20		20	40
Tot	80	45	10	70	35	30	20	55
	40	40	40	40	40	40	40	40
Δ%	-40	-5	30	-30	5	10	20	-15

Capacità richiesta → (Tot)
" disponibile → (40)

Supponendo di usare t, b = 1 sett, 1 turno lavorativo al giorno (quindi 40h/sett)

42

Purtroppo però, l'RCCP va un po' troppo "a spanne", in quanto:

- non si tiene conto di come il lavoro sia schedulato all'interno del centro di lavoro (non si considerano ad es. gli idle times)
- non è presente una fase di netting (si usa sempre FL) e ciò, sebbene possa essere accettabile per i prodotti finiti, non lo è per i semiassemblati o per i componenti

fornisce quindi, spesso, una stima un po' troppo ottimistica ed è inoltre meno dettagliato del CRP, un altro strumento x effettuare checks sulla capacità

- <u>Capacity Requirements Planning (CRP)</u>: fornisce un più dettagliato check sulla capacità sulla base dei piani di produzione generati dall'MRP

è una sorta di MRP sulla capacità: usa le info sul routing del processo e sull'MRP per valutare meglio la capacità (ovviamente da valutare x ogni singolo centro di lavoro)

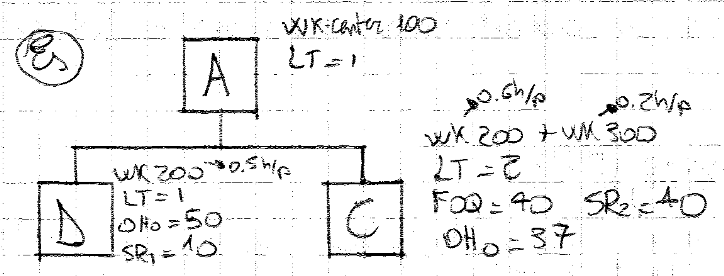
richiede in input tutti gli OR, i livelli di WIP esistenti, le info sul routing, la capacità e i LT per ogni centro di processo

purtroppo però, essendo una sorta di MRP, stima i tempi di completamento dei JOB usando LT fissi (assumendo quindi la capacità infinita) per poi calcolare i cosiddetti loading over time.

tali loadings sono quindi comparati alla capacità disponibile ma nessuna correzione viene effettuata in caso di situazioni di overload

Per tale motivo CRP NON viene utilizzato!

Es



MRP di A

t	1	2	3	4	5	6	7	8	9	10
OR	33	33	33	40	40	40	30	30	30	37

diventano il FL di C che quindi avrà un piano

MRP di C

OR	40	40	40	40	40	40		40		

Trasformiamo ora le qtà in ore e calcoliamo CRP per WK 300 $(0{,}2h \cdot 40 = 8h)$

t	1	2	3	4	5	6	7	8	9	10
C(h)	8	8	8	8	8	8		8		

Quando vi sono più items di alto livello che usano le medesime parti di livello più basso, sorge un conflitto a causa della quantità di on-hand disponibile insufficiente

↓

allocando le parti ad un job piuttosto che ad un altro, la funzione job release può razionalizzare tale conflitto

↓

alloca, difatti, le parti disponibili ai job che conviene soddisfare ed emette una notifica che avvisa il sistema della mancanza di on-hand (le notifica resta nel job pool sino a quando la quantità richiesta non possa essere rilasciata)

↓

Una volta che un job/purchase order è stato rilasciato, bisogna effettuare alcuni controlli x assicurarsi che venga completato per tempo con le corrette quantità e specifiche:
- se il job è per componenti da acquistare, si deve tracciare il purchase order (in modo da poterlo monitorare e sapere se e quando arriva)
- se il job è per la produzione interna, cade sotto la funzione detta <u>Shop floor control (SFC)</u> o <u>Production activity control (PAC)</u>

↓

Le 2 funzioni principali dell'SFC sono:

• <u>Job dispatching</u>: è una parte della schedulazione, la cui idea base è quella di sviluppare una regola per organizzare la coda davanti ad ogni workstation allo scopo di preservare l'integrità delle due dates dei vari job, mantenendo alto l'utilizzo delle macchine e bassi i tempi di produzione

↓

tali regole sono dei semplici metodi euristici, analoghi alle regole di lot sizing, che analizzeremo quando tratteremo la schedulazione (es. SPT, EDD, Least Slack, Critical Ratio...)

• <u>Input/Output Control</u>: è un metodo per mantenere i lead times sotto controllo:

- si monitora il livello di WIP in ogni centro di lavoro
- se WIP supera un determinato livello, il tasso di rilascio è troppo elevato e quindi dovrò ridurlo
- se WIP scende sotto un determinato valore, il tasso di rilascio è troppo basso e bisogna quindi aumentarlo
- se WIP è compreso tra i due valori sopracitati, significa che il tasso di rilascio è adeguato alle condizioni correnti

(il tasso di rilascio viene aumentato o diminuito apportando dei cambiamenti all'MPS)

I/O Control fornisce quindi un modo semplice per fare un check sui rilasci sulla base della capacità disponibile. Tuttavia aspettando che il WIP raggiunga livelli eccessivi prima di intervenire si rischia che il sistema sia già fuori controllo. Questo è uno dei motivi per cui i cosiddetti SISTEMI PULL (tipo il

Cap. 11: Variability and its impact on Process Performance: Waiting Time Problems

Dal punto di vista dei clienti la forma più evidente di mismatch tra domanda e offerta è data dai TEMPI D'ATTESA

è importante distinguere tra 2 tipi di tempi d'attesa (waiting times):

- quelli causati da problemi di capacità, dove Utilizzo Implied > 100%

  in questi casi si ha un'ovvia presenza di code, e quindi di tempi d'attesa, in quanto il tasso di domanda eccede il tasso di produzione per alcuni periodi di tempo

  ciò capita prevalentemente in casi di capacità limitata e tasso di domanda affetto da stagionalità (*)

  essendo un problema di capacità, la variabilità è solo un effetto secondario (dato che è tutto deterministico e costante)

  ← per cui tentare di ridurre la variabilità non risolve il problema

- quelli causati da variabilità, dove si può anche avere Utilizzo Implied < 100%

  in presenza di variabilità, possono crearsi code, e quindi tempi d'attesa, anche nei casi in cui il processo è demand-constrained (ossia I-U < 100%), ossia anche quando vi è abbastanza capacità, IN MEDIA, per soddisfare la domanda

  in tali casi è fondamentale cercare di ridurre al minimo tale variabilità per, quanto meno, limitare il problema (temporanee insufficienze di capacità)

→ Con variabilità si intende quella indotta dalla casualità che non è trascurabile né deterministicamente nota

per tale motivo i tempi d'attesa da essa generati non sono facilm. prevedibili a priori

Obiettivo del management è quindi quello di:

- prevedere i tempi d'attesa e derivare delle misure di performance che colgano la qualità del servizio fornito al cliente
- individuare delle vie per ridurre i tempi d'attesa scegliendo appropriati livelli di capacità, ridisegnando il sistema e delineando opportunità per ridurre la variabilità

(*) OSS. Stagionalità è una forma di NON stazionarietà ma NON stazionarietà $\not\Rightarrow$ stagionalità (un processo non stazionario non per forza è stagionale)

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 91 di 229

difatti, - buffer di MG : inammissibile xk non posso avere, in anticipo, un gruppo di persone che sanno che avranno sicuram. bisogno di un intervento d'emergenza

- buffer di tempo : se è un'emergenza non posso far aspettare

Ⓒ Trapianti di organi:

unico buffer ammissibile è quello di tempo : liste d'attesa

- buffer di MG : fisicamente inammissibile in quanto non posso tenere in MG degli organi dato che sono altamente deperibili

- buffer di capacità : eticamente inammissibile in quanto dovrei creare un MG di gente da ammazzare in caso di necessità

⇒ la variabilità esiste, pertanto se non voglio patirne le conseguenze (costi) dovrò attrezzarmi in anticipo scegliendo come bufferizzare

se più di un tipo di buffer è ammissibile, sceglierò quello di costo minore

Detto questo, minore è la variabilità in gioco, minore sarà la necessità di buffer, pertanto si rende necessario studiare la variabilità per capire:

- da dove arriva
- come misurarla
- come ridurla

- Perchè esiste la variabilità nei sistemi ?

Per quanto riguarda la radice ultima della casualità (variabilità) vi sono 2 correnti di pensiero principali:

- Einstein : "DIO NON GIOCA A DADI" → la casualità non esiste ma è solo conseguenza del fatto che noi non conosciamo quella data cosa/fenomeno

- Bohr : "NON dire a Dio cosa deve fare" → casualità esiste ed è intrinseca nell'universo (essere umano non può capire tutto)

Ma da un punto di vista manageriale, e non filosofico, quello che realmente ci interessa non è la radice ultima della casualità, bensì la fonte :

Sorgenti di Variabilità :

① Processi di arrivo (domanda)
② Tempi di processo (o di servizio)
③ Risorse
④ Routing

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 93 di 229

- Come misurare la variabilità?

In generale, ogni forma di variabilità è misurata con la deviazione standard ($\sigma^2 = E[X^2] - (E[X])^2$)

↳ la deviazione standard è una misura ASSOLUTA della variabilità che dà informazioni riguardo lo scostamento dalla media

Però per poter dire se tale valore è alto o basso, buono o da migliorare, bisogna rapportarlo al tempo di processo

Necessitiamo, cioè, di una misura RELATIVA della variabilità.
Definiamo pertanto il:

$$\boxed{\text{Coefficiente di Variazione} \quad CV = \frac{\text{deviazione standard}}{\text{media}}}$$

unico modo x poter confrontare la mia azienda con un'altra o per far confronti internamente all'azienda, ad es. per verificare se dei cambiamenti abbiano prodotto gli effetti sperati

OSS. Dal momento che la deviazione standard e la media hanno le stesse unità di misura, il coefficiente di variazione sarà adimensionale

Processi di Arrivo:

Quando analizziamo problemi con variabilità, è di particolare importanza un'accurata rappresentazione della domanda, che determina i tempi di arrivo dei clienti.

Iniziamo col definire:

- Tempo di Arrivo del cliente i $= AT_i$
- Tempo di interarrivo $= IA =$ tempo tra 2 arrivi consecutivi (quindi $IA_i = AT_{i+1} - AT_i$)

↳ variabile casuale

↳ facendo la media dei vari IA di un determinato intervallo di tempo, si ottiene l'ampiezza dell'intervallo di tempo in cui in media arriva 1 cliente

es call center: dire "in media arriva 1 telefonata ogni 5 minuti", quei "5 minuti" rappresentano la media degli interarrivi (a

L'analisi di un processo di arrivo deve cominciare con le seguenti 2 domande:

① Il processo di arrivo è STAZIONARIO?

② I tempi di interarrivo sono distribuiti ESPONENZIALMENTE in modo da formare un processo di arrivo Poissoniano?

Un processo di arrivo è detto Poissoniano se ha tempi di interarrivo esponenzialmente distribuiti

Il fatto che il processo di arrivo sia Poissoniano è particolarmente importante a causa di due importanti proprietà della distribuzione esponenziale:

- deviazione standard ≡ media (tempo medio di interarrivo a) $\Rightarrow CV = 1$
- variabile esponenziale (tempi di interarrivo) è SENZA MEMORIA

il numero di arrivi in un intervallo di tempo è indipendente da quando è avvenuto l'ultimo arrivo

ossia il valor medio non dipende dal tempo in cui è avvenuto l'evento precedente (l'ultimo evento)

l'arrivo di un cliente non mi dà nessuna info su quando arriverà il prossimo

gli arrivi (clienti) sono INDIPENDENTI l'uno dall'altro, non c'è CORRELAZIONE

ciò comporta un forte alleggerimento dal punto di vista analitico

Se i tempi di interarrivo non fossero esponenzialmente distribuiti, avremmo bisogno di più parametri per descrivere il processo d'arrivo

Nel caso di distribuzione esponenziale, invece, il tempo medio di interarrivo è uguale alla deviazione standard dei tempi di interarrivo, pertanto è sufficiente un solo parametro per caratterizzare l'intero processo di arrivo.

$\Rightarrow$ Se la distribuzione è esponenziale $CV_a = 1$

Per determinare se i tempi di interarrivo sono esponenzialmente distribuiti:

1. Ordinare i tempi di interarrivo in ordine crescente $(a_i \dots a_n)$
2. Tracciare la funzione empirica di distribuzione (mettendo a_i sulle ascisse e i/n sulle ordinate)
3. Confrontare il grafico così ottenuto con una distribuzione esponenziale con un parametro scelto in modo appropriato (parametro $= a$ in genere è appropriato)

se la distribuzione assomiglia ad una esponenziale devo procedere con un fitting (interpolazione) per poi cercare media e deviazione standard

OSS.
- Se processo NON è stazionario → NON posso fare nulla (devo cercare di stazionarizzarlo)
- Se processo NON è poissoniano → posso solo fare una simulazione (analizzando il sistema e la sua evoluzione nel tempo)