



SCM200

Lager- og produksjonsplanlegging

Kompendium 2019, del 2

Bård-Inge Pettersen
Per Kristian Rekdal



Høgskolen i Molde
Vitenskapelig høyskole i logistikk

Innhold

1	Grunnleggende Begreper	9
1.1	Målsettinger	10
1.2	Planlegging og styring	11
1.3	Beslutningshierarkiet	14
1.4	Planleggings- og styringssystemer MPC	19
1.5	Modulene i et MPC-system - Verdensbildet	22
1.6	Modulene i et MPC-system	24
1.6.1	Strategiske moduler i et MPC-system	24
1.6.2	Taktiske moduler i et MPC-system	26
1.6.3	Operasjonelle moduler i et MPC-system	28
1.7	Klassifisering av bedrifter	29
1.8	Kundeordrepunkt	32
1.9	MTS, ATO, MTO, ETO - klassifisering	36
1.10	Klassifisering av produksjonsprosesser	38
2	Demand Management	41
2.1	Oversikt	42
2.1.1	Verdensbildet (globalt)	43
2.2	Demand Management - hovedaktiviteter	44
2.2.1	Kundeordrepunkt - aktiviteter	47
2.3	Prognostisering	49
2.3.1	To typer prognosemetoder	52
2.4	Mønster	56
2.5	Motivasjon: Prognostisering av tilfeldig mønster	59
2.6	Generelt: Prognostisering av tilfeldig mønster	63
2.7	Metoder for prognostisering av tilfeldig mønster	67
2.7.1	Random Walk, RW	69
2.7.2	Glidende gjennomsnitt $MA(n)$	75
2.7.3	Vektet glidende gjennomsnitt med vindu n , $WMA(\theta_t, n)$	81
2.7.4	Eksponensiell glatting $ES(\theta)$	91
2.8	Måltall	105
3	Strategisk Planlegging	121
3.1	Strategisk Planlegging	122
3.2	Strategisk Ressursplanlegging	124
3.3	“Strategisk ressursplanlegging” - beslutninger og hovedaktiviteter	125
3.3.1	Beslutninger	125

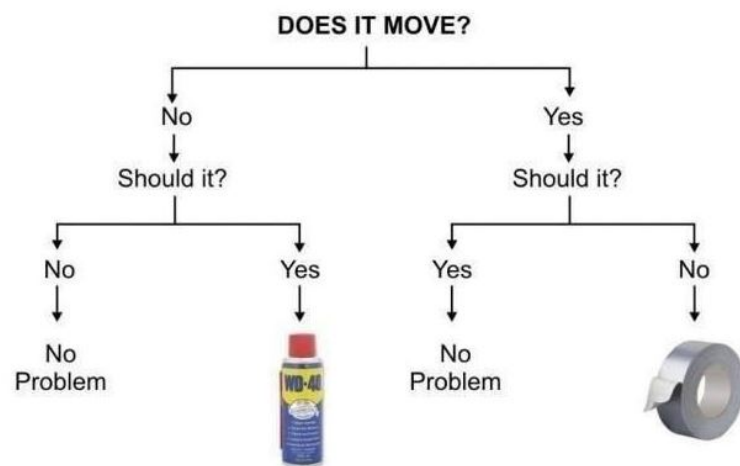
3.3.2	Effektivitet og kapasitetsutnyttelse	125
3.4	Fasilitetsdesign	130
3.5	Målsetting	133
3.6	Varianter	134
3.7	Matematisk Modellering	137
4	Aggregert Planlegging	147
4.1	Oversikt: Aggregert Planlegging	148
4.1.1	Verdensbildet - globalt	150
4.1.2	Verdensbildet - lokalt	151
4.2	Aggregert planlegging - målsettinger	152
4.3	Aggregert planlegging - beslutninger	153
4.4	Aggregert planlegging - hovedaktiviteter	154
4.5	Ressursplanlegging	155
4.6	Aggregert planlegging: Glamox	156
4.6.1	Grunnleggende tilfelle (situasjon 0)	158
4.6.2	Utvidelse 1: Sikkerhetslager	174
4.6.3	Utvidelse 2: Mulighet for bestilling	180
5	Master Produksjonsplanlegging	191
5.1	Master produksjonsplanlegging: Innledning	192
5.2	Master produksjonsplanlegging - Hovedaktiviteter	194
5.2.1	Typiske varer i MPS	195
5.2.2	Typiske aktiviteter i MPS	196
5.2.3	Typiske poster i MPS	197
5.3	Lagerstyring	203
5.4	EOQ - Optimal Ordrestørrelse ved konstant etterspørsel og ingen stockout	210
5.5	Etterbestilling - dvs. når legger vi inn nye ordrer	219
5.5.1	Bestilingspunktmodellen - uten og med sikkerhetslager	220
5.6	Fastsettelse av sikkerhetslager under krav om sannsynlighet for stockout	229
5.7	Varierende etterspørsel - Wagner-Within	238
5.7.1	Matematisk modell - uten kapasitetsføringer	239
5.7.2	Wagner-Within algoritmen	254
6	MRP - Material Requirements Planning	281
6.1	Introduksjon	282
6.2	Hovedaktiviteter	285
6.3	BOM-strukturer	287
6.4	MRP-modellen	289
6.5	MRP-tabellen	301
6.6	MRP-algoritmen (planleggingsmetodikken)	303
6.7	ABC-analyse	318
7	Operasjonell planlegging	323
7.1	Introduksjon	324
7.2	Jobb allokering	327
7.3	Sekvensiering	328
7.3.1	Målfunksjoner	329

7.3.2	Gantt-diagrammer	332
7.4	Sekvensieringscase: En stasjon med 6 ordrer	333
7.5	Sekvensieringscase: Sykehus	340
7.6	Sekvensieringscase: To maskiner	344

Kapittel 5

Master Produksjonsplanlegging

Engineering Flowchart



5.1 Master produksjonsplanlegging: Innledning

I forrige kapittel, dvs. kap. 4, gikk vi gjennom [aggregert produksjonsplanlegging](#), hvor vi samlet familier av produkter basert på forskjellige aggregeringsfaktorer.

En aggregert familie kan [disaggregeres](#) til de enkelte produktene, slik at summen av enkeltproduktene i familien utgjør den totale familien. Vi får da et sett med ettersørselsbehov for de [faktiske](#) produktene.

Hvis den aggregerte planleggingsmodulen ikke benyttes, skaffer vi inn etterspørselsbehovene fra prognoser/ordrer fra Demand Management modulen (se gul pil i figur 5.2).

[Master produksjonsplanlegging](#) tar imot de disaggregerte behovene for hver enkelt produkt og planlegger hvordan disse produktene skal kjøpes inn/produseres.

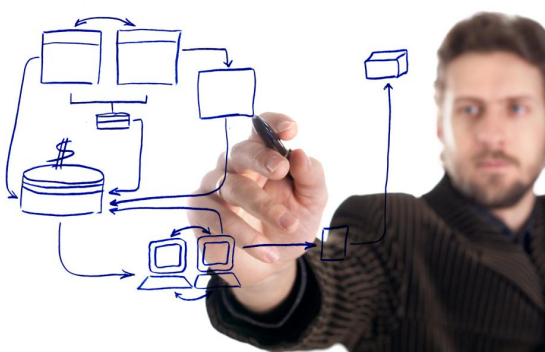


Foto: Colourbox

Figur 5.1: Master produksjonsplanlegging.

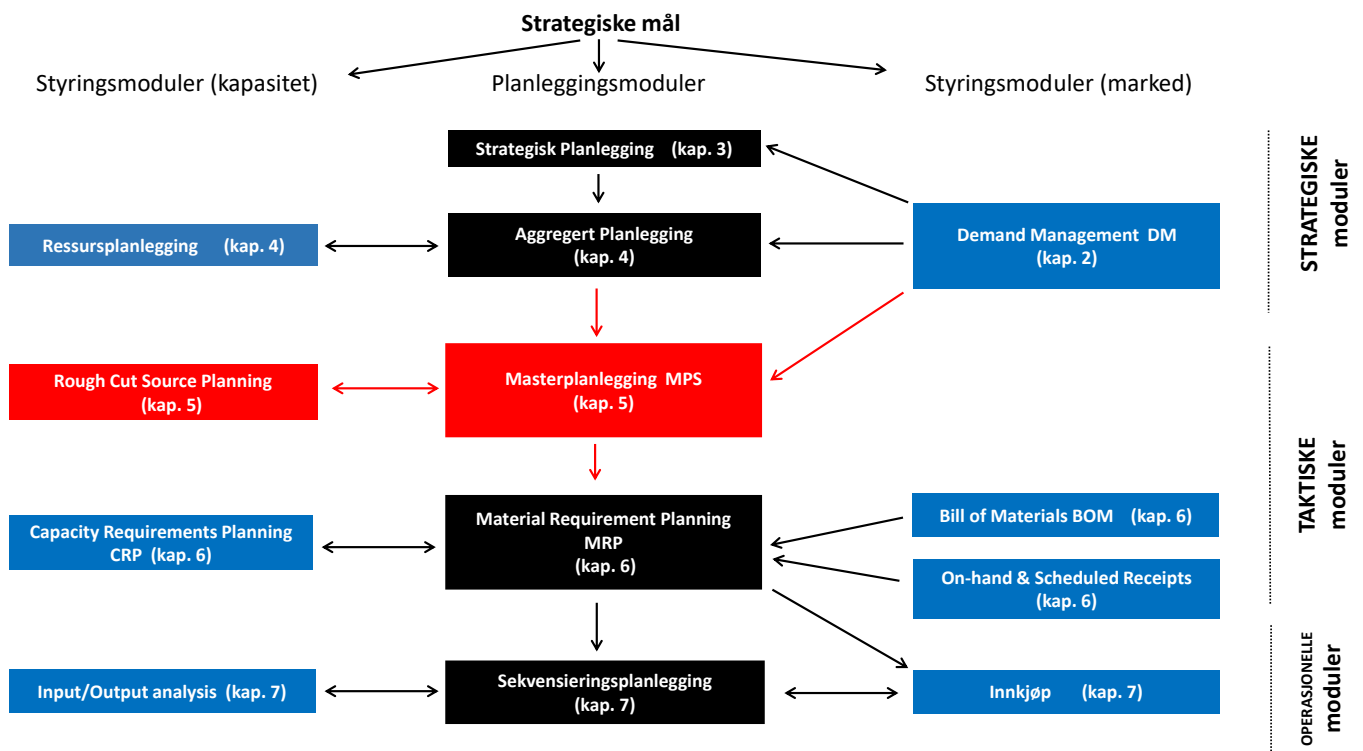
Verdensbildet: (globalt)

Master produksjonsplanleggingsmodulen (MPS) er en taktisk modul som blant annet får data fra

- *strategisk ressursplanlegging*
- *Demand Managament* (DM)
- *aggregert produksjonsplanlegging* (MPS)

se gule piler i figur 5.2.

Master produksjonsplanlegging (MPS) blir deretter input til *Material Requirement Planning* (MRP) modulen.



Figur 5.2: Master produksjonsplanlegging (MPS).**NY FIGUR**

5.2 Master produksjonsplanlegging - Hovedaktiviteter

Master produksjonsplanlegging svarer på følgende **hovedbeslutninger**:

- *hvilke* varer holder vi masterplan for?
- *hvor mye* av hver vare skal vi anskaffe?
- *hvor mye* av hver vare skal vi ha på lager?
- *når* skal varene være på lager?

Modulen har også noen viktige **beslutninger** ved styring av *master produksjonsplanen* (MPS): (gjennomføring av planen):

- hvilke perioder er *fryste*, dvs perioder hvor vi følger faktiske ordrer fremfor prognose?
- hvilke perioder er *fleksible*, dvs hvor vi følger den største verdien av faktiske ordrer og prognose?
- hvilke perioder er *frie*, dvs hvor vi følger kun prognose?
- gitt en ny ordre, skal vi godta ordren eller ikke?

5.2.1 Typiske varer i MPS

Generelt kan vi si at vi holder MPS for varer som utgjør **uavhengig etterspørsel**. Vi husker at uavhengig etterspørsel er etterspørsel som ikke er avhengig av selve bedriftens produktstruktur.

- Uavhengig etterspørsel er et resultat av kundens ønsker.
- Avhengig etterspørsel er derimot helt avhengig av hvordan produktstrukturen er oppdelt i videre komponenter, og er følgelig noe bedriften selv har kontroll på.

Kundeordrepunktet, se side 34, er det punktet i bedriften hvor etterspørsel går fra uavhengig til avhengig etterspørsel. Vi husker at vi har noen typiske kundeordrepunkt:

- **Make-To-Stock**, holder lager av ferdigvarer, så masterplanen (MPS) omhandler ferdigvarer på lager.
- **Assemble-To-Order**, holder lager av halvfabrikater, så masterplanen (MPS) omhandler halvfabrikater på lager.
- **Make-To-Order/Engineering**, holder lager av råvarer og komponenter, så masterplanen (MPS) omhandler råvarer og komponenter på lager.

5.2.2 Typiske aktiviteter i MPS

Typiske planleggingsaktiviteter:

- innhente disaggregerte **behovsprognoser** fra *aggregert produksjonsplan* eller prognoser fra *Demand Management* (DM)
- hvor lang skal **planleggingshorisonten** være, typisk fra 4 - 16 uker
- bestemme hvilke perioder som er fryste, fleksible og frie
- bestemme hvor mye skal vi **produsere/bestille** og hvor mye skal vi ha på **lager** for hver periode
- bestemme hvordan etterspørsel skal oppfylles, dvs. om vi skal tillate stockout eller ikke
- bestemme **servicegrad**
- bestemme hvorvidt vi skal holde sikkerhetslager, og hvor stort det eventuelt skal være

Typiske styringsaktiviteter:

- **oppdatere** MPS etterhvert som salgstall tikker inn. Vi sier gjerne at ordrene ”spiser” fra prognosen.
- bestemme når MPS skal **replanlegges**, som en eventuell følge av større endringer
- avgjøre om eventuelle nye ordrer kan **leveres** eller ikke

5.2.3 Typiske poster i MPS

Materplanen (MPS) er en såkalt **rullerende** plan.

Det vil si at planen har en gitt **horisont**, si f.eks. 10 uker, og etter hver uke går sist uke ut av planen, mens en ny uke kommer inn.

En masterplan består av følgende **element** (for hver periode):

- **Prognose** - de prognostiserte verdiene i løpet av horisonten
- **Ordrer** - bekreftede ordrer som skal leveres i løpet av horisonten
- **Lager** - planlagte lagernivåer i løpet av horisonten
- **Ledig-til-reservasjon** - reflekterer hvor mye vi har lovet bort fra en gitt bestilling/produksjon. Et styringstall som gir en rask måte å se om en ordre kan leveres eller ikke for en gitt periode.
- **Master Produksjon Plan** (MPP) - produksjons- eller innkjøpsplaner. Tallene viser når varene skal ankomme, ikke når de bestilles.

Uke t	init	1	2	3	4	5	6	7	8
Prognose F_t									
Ordrer D_t									
Lagernivå I_t									
Ledig-til-reservasjon R_t									
MPS									

Tabell 5.1: Masterplan (MPS).

Eksempel: (masterplan, MPS)

Anta vi jobber som planleggere ved Elkjøp sitt hovedlager, og vi skal lage en masterplan (MPS) for et av deres viktigste produkter: iPhone 6. Anta vi har oppgitt følgende data:

- Ledetid: 2 uker
- Fast ordrestørrelse: $MPS = 6000$ iPhoner
- Vi bruker en **planleggingspolicy** om at vi legger inn en ny bestilling hver gang vi ser vi blir tom på lager
- Initielt lager er $I_0 = 5600$ iPhoner
- Fryst sone er 2 uker, fleksibel sone er 6 uker og fri sone er 2 uker

Masterplanen (MPS) for iPhone de neste 8 ukene blir:

Uke t	init	1	2	3	4	5	6	7	8
Prognose F_t		2200	2500	2000	2000	1800	1800	3200	2000
Ordrer D_t		2400	2300	1700	2200	1500	1400	1700	2200
Lagernivå I_t	5600	3200	900	4900	2700	900	5100	1900	5900
Ledig-til-reservasjon R_t		900		600			2900		3800
MPS				6000			6000		6000

Tabell 5.2: Masterplan (MPS).



Figur 5.3: Elkjøp og iPhone.

- a) Bruk lagerbalanse og finn lageret for uke $t = 3$, dvs. finn I_3 .
- b) Hva er *ledig-til-reservasjon* i første uke $t = 1$, dvs. hva er R_1 ?
- c) Hva er *ledig-til-reservasjon* i tredje uke $t = 3$, dvs. hva er R_3 ?
- d) Det kommer inn en ny ordre på 800 iPhoner i uke $t = 2$.
Kan vi *garantere* kunden at vi kan levere denne ordren?
- e) Anta videre at det kommer inn en ny ordre i uke $t = 4$ på 1000 iPhoner.
Kan vi *garantere* kunden at vi kan levere denne ordren?
Hvis ikke, hva slags alternativer kan du gi til kunden?

Løsning:

a) Fra lign.(4.56) på side 170 vet vi at lagerbalansen for hver uke er:

$$\text{inngående lager} + \text{inn til lager} - \text{ut av lager} = \text{utgående lager} \quad (5.1)$$

som gir

$$\underbrace{I_2}_{= 900} + \underbrace{MPS}_{= 6000} - \underbrace{D}_{= 2000} = I_3 \quad (\text{lagerbalansen i } t = 3) \quad (5.2)$$

$$900 + 6000 - 2000 = I_3 \quad (5.3)$$

dvs. utgående lager I_3 er:

$$\underline{\underline{I_3 = 4900}} \quad (5.4)$$

- b) Vi har *alltid* en verdi for *ledig-til-reservasjon* i *første* uke, R_1 , siden denne representerer det vi hadde igjen fra foregående uke, dvs. initielt lager $I_0 = 5600$.
Beregningen for *ledig-til-reservasjon* i første uke, dvs. R_1 , er:

$$\underline{\underline{R_1}} = I_0 - D_1 - D_2 \quad (5.5)$$

$$= 5600 - 2400 - 2300 = \underline{\underline{900}} \quad (5.6)$$

Hvis vi hadde hatt en bestilling i første uke, måtte vi lagt til denne.

- c) Legg merke til at *ledig-til-reservasjon* svarer alltid til en MPS-verdi, siden vi starter å selge fra en planlagt bestilling *før* den har kommet. Vi trekker ifra alle inneværende ordrer *neste* MPS-verdi. *Ledig-til-reservasjon* for uke $t = 3$, dvs. R_3 , er dermed:

$$\underline{\underline{R_3}} = MPS - D_3 - D_4 - D_5 \quad (5.7)$$

$$= 6000 - 1700 - 2200 - 1500 = \underline{\underline{600}} \quad (5.8)$$

- d) Vi ser på *ledig-til-reservasjon* $R_3 = 900$ dekker denne perioden, som er 900. Derfor: Ja, vi kan garantere å levere 800 iPhoner til kunde.

- e) Siden *ledig-til-reservasjon* er $R_3 = 600$ så kan vi dessverre ikke garantere leveransen for ordren på 1000 iPhoner.
Men vi har nå hjelpemidlet for å gi kunden alternative løsninger:

- 1) Vi splitte ordren i to leveranser. Gi kunden $R_3 = 600$ iPhoner med en gang. Og de resterende 400 iPhonene i uke $t = 6$.
- 2) Alternativt kan vi levere alle 1000 iPhonene i uke $t = 6$.

■

5.3 Lagerstyring

Siden masterplanlegging innebærer faktisk lagring (i motsetning til lageret beregnet ved aggregert planlgging, som beregner sammenslåtte lager), passer det nå å se nærmere på [lagerstyring](#) generelt.

Vi påpeker at alt som beskrives i dette kapittelet også er gjelder for [MRP-modulen](#) i kapittel 6, men da for [komponenter](#).

[Lager](#) er en nødvendig funksjon i en bedrift som er der for blant annet å :

- Forbedre [leveringsgraden](#) til kundene
- Være en buffer for [variasjon](#) og [usikkerhet](#) i hele verdikjeden, fra kundesalg til ledetider til uforutsette hendelser.
- Kunne gi mer effektiv produksjon gjennom økte [seriestørrelser](#)

Men lager har også en betydelig [ulempe](#): [lagerkostnader](#)



Figur 5.4: Lager.

Lagerkostnader:

Lagerkostnaden er en **komplisert og sammensatt** kostnad, som er ofte krevende for bedriften å få fremskaffet pålitelige data. Eksempler på typiske lagerkostnader er:

- **Lagerplass/fasiliteter** - en drifts -og investeringskostnad for å kunne ha lager
- **Forsikring** - forsikringskostnader på selve lagervarene
- **Skatter og avgifter** - lager beregnes som verdier som inngår i skatteoppgjøret
- **Kapitalkostnader** - penger investert i varelager kan ikke samtidig brukes til andre formål. Beregnes som regel gjennom det statlige **rentenivået**.
- **Inkurans** - varer går ut på dato eller går av moten
- **Svinn** - tyveri eller ødeleggelse av lagervarer
- **Administrasjonskostnader** - ansatte som jobber med lageret

Vi skal i dette kurset nesten utelukkende anta at lagerkostnaden kun består av **kapitalkostnaden**, dvs rentenivået. Men i virkeligheten er kostnaden mye mer nyansert.

Men å **ikke** å ha varen på lager når kunden ønsker varen, er også forbundet med store kostnader - denne situasjonen kalles **stockout** og kostnaden kalles følgelig **stockout-kostnaden**.



Foto: Colourbox

Figur 5.5: Lagerkostnader.

Kostnader ved stockout:

Stockoutkostnaden er kanskje en enda mer **komplisert og sammensatt** kostnad enn lagerkostnaden, siden begrepet **kundetilfredshet** er vanskelig å sette en kostnad på. Eksempler på kostnader assosiert med stockout er:

- **Dårlig kundeservice** - ingen kunder like å måtte bestille varen og dermed vente til de faktisk får varen. Det kan i verste fall medføre at kunden skifter leverandør.
- **Bestillingskostnader** - dersom det forekommer stockout, må bedriften ofte legge inn hastbestillinger, som dermed medfører **økte** bestillingskostnader eller omstillingskostnader (dersom produksjon finner sted).
- **Etterlevering** - ordrer som splittes opp må etterleveres, noe som medfører økte transportkostnader.
- **Produksjonsproblemer** - produksjonen er ikke **synkronisert** med markedet, noe som medfører at bedriften ikke klarer å respondere raskt ved svingninger i markedet og dermed svekkes konkurranseevnen.
- **Lagerutnyttelse** - investerte lagerfasiliteter står mye tomme ved stockout, noe som innebærer dårlig utnyttelse av lagerfasilitetene.

Lagerkostnaden og stockoutkostnaden er to **motstridende** kostnader - dvs hvis den ene er høy er den andre ofte lav og vice versa. Det oppstår dermed nesten *alltid* et behov for å minimere disse kostnadene under ett. Masterplanlegging har ofte dette som hovedmål.

Men hvilke **typer** lager har vi? Det skal vi se på i neste avsnitt.



Figur 5.6: Stockout.

Lagerkategorier: ([hvordan lager oppstår](#))

Lager oppstår i flere fasonger. Her er noen av dem:

- [Transittlager](#) - lager under prosess. Dette er lager som oppstår i flere situasjoner, for eksempel varer i arbeid eller varer under transport.
- [Sykluslager](#) - Lager som veksler mellom høyt nivå og lavt nivå . Lageret har som hensikt å gi fordeler som f.eks. serieproduksjon eller rabatterte innkjøp.
- [Sesonglager](#) - Lager som tar høyde for sesongvariasjoner.
- [Sikkerhetslager](#) - lager som tar høyde for usikkerheten i markedet/leverandører.
- [Frikoblingslager](#) - lager plassert mellom to påfølgende operasjoner i en linjeproduksjon. Hensikten er å kunne frikoble den ene stasjonen fra den andre, med mål om mer effektiv bruk av maskiner og arbeidskraft.
- [Reservelager](#) - lager som holder reservedeler.

Nå som vi har sett på forskjellige lagerkategorier, spør vi oss: hvordan [styres](#) lageret? Det er tema for neste avsnitt.

Neste spørsmål er hvordan vi bestemmer [seriestørrelsene/ordrestørrelsene](#) i masterplanen, dvs. MPS-verdiene i masterplantabellen. Seriestørrelse reflekterer at vi har produksjon, mens ordrestørrelse representerer et rent lagerssystem. Dette diskuteres i neste avsnitt.



Figur 5.7: Lagerkategorier.

Seriestørrelser: (hvordan bestemmer vi hvilken seriestørrelse/ordrestørrelse vi skal ha?)

I eksempelet på side 198 som viste en masterplan for Elkjøps iPhone, hadde vi at **ordrestørrelsen** var allerede oppgitt til 6000 iPhoner som vi bestilte etter behov. Hvor fikk vi 6000 ifra?

Ordrestørrelsen er kanskje den **viktigste beslutningen** vi tar når det gjelder masterplan (og også MRP-plan som vi ser på i neste kapittel).

Vi har følgende metoder/modeller vi bruker for å bestemme ordrestørrelser/seriestørrelser, hvor vi skiller mellom tilfellet om vi har **konstant etterspørsel** eller ikke.



Figur 5.8: Serieproduksjon.

A) Enkle metoder:

Enkle metoder innebærer enten store forenklinger eller at de ikke tar høyde for lagerkostnader og bestillingskostnader/setupkostnader.

1. Konstant etterspørsel

- (a) **fast valgt ordrestørrelse** - vi bestiller/produserer samme kvantum hver gang. Metoden er kun egnet for tilfeller med konstant etterspørsel.
- (b) **EOQ - Economic Order Quantity** - en eksplisitt formel for **optimal** ordrestørrelse, under flere relative strenge forutsetninger, hvor den viktigste er konstant etterspørsel. Metoden er mye brukt på grunn av dens robusthet og enkelhet.

2. Varierende etterspørsel

- (a) **Lot-For-Lot** - vi bestiller/produserer akkurat det vi har behov for i den perioden vi ser på. Metoden passer for **kostbare** varer, siden metoden minimerer lagernivået. Siden Lot-For-Lot minimerer lageret, økes bestillingskostnaden/setupkostnadene betraktelig. Metoden passer derfor for multitask maskiner som raskt kan bytte produkt slik at setup-kostnaden blir neglisjerbar.
- (b) **FPR - Fixed Period Requirements** - vi bestiller inn akkurat det vi trenger for et gitt antall perioder. Metoden passer fint dersom vi har lave lagerkostnader i forhold til bestillingskostnaden/setupkostnaden.
- (c) **POQ - Period Order Quantity** - nesten det samme som FPR - forskjellen er at vi bruker EOQ for å beregne hvor mange periodebehov vi skal inkludere.

B) Avanserte metoder:

Avanserte metoder innebærer at vi *tar* høyde for merkostnadene som lagerkostnader og bestillingskostnader/setupkostnader. Alle metodene håndterer varierende etterspørsel:

1. **Wagner-Within metode** - matematisk optimeringsmodell som gir optimale kostnader.
2. **LUC - Least Unit Cost** - vurderer et gitt sett av ordrestørrelser ved å beregne enhetskostnaden for hver ordrestørrelse. Vi velger den ordrestørrelsen med lavest enhetskostnad.
3. **LTC - Least Total Cost** - vurderer også et gitt sett av ordrestørrelser, men beregner den **totale merkostnaden** og velger seriestørrelsen med lavest total merkostnad.
4. **Silver-Meal heuristic** - en rask tilnærming til optimal løsning, som likner litt på LUC-metoden.

Av alle disse metodene, skal vi se nærmere på *to* av dem: **EOQ** og **Wagner-Withins metode**. Disse representerer to ytterligheter, og de er begge svært fundamentale metoder i produksjonsplanlgging. Vi starter i neste avsnitt med EOQ-modellen.

5.4 EOQ - Optimal Ordrestørrelse ved konstant etterspørsel og ingen stockout

EOQ eller **Economical Order Quantity** er en svært utbredt og populær metode for å beregne ordrestørrelse (både produksjon og innkjøp). Årsakene til dette er flere:

- **Robust** - metoden er robust, dvs til tross for svært strenge forutsetninger, så vil metoden gi gode tilnærmelser selv om en eller flere av antakelsene er brutt.
- **Minimerer kostnader** - metoden veier opp bestillingskostnader/setupkostnader mot gjennomsnittlige lagerkostnader og søker å minimere de totale kostnadene.
- **Enkel** - metoden er enkel å *implementere* i bedriften, dvs det er enkelt å få metoden i bruk rent praktisk.
- **Utgangspunkt** - metoden er ofte et utgangspunkt for andre mer kompliserte metoder



Figur 5.9: EOQ.

EOQ: (forutsetninger og formel)

EOQ har følgende forutsetninger:

- **Konstant og kontinuerlig** etterspørsel - dette er den klart **strengeste** forutsetningen
- **Inge stockout** - vi antar at etterspørselen oppfylles
- **Uendelig prosess** - vi antar at etterspørselen hverken har en start eller slutt
- **Inge føringer** - vi antar at det ikke finnes noen føringer på ordrestørrelsen (for eksempel lagerkapasitet, produksjonskapasitet, bestillingskapasitet, tilgjengelig kapital etc)
- **Uendrede kostnader** - kostnadene (lagerkostnader og ordrekostnader) i modellen er konstante, dvs. de endres ikke over tiden
- **Ingen rabatter** - vi antar at økte ordrestørrelser ikke gir kvantumsfordeler som for eksempel rabatter eller kvantumsfordeler ved produksjonen.
- **Total leveranse** - vi antar at alle enhetene som ble bestilt ankommer lageret i en batch. Dette er ikke alltid tilfelle, f.eks. ved maskiner som har en gitt produksjonsrate (f.eks. Hustadmar-mor)

Setning: (EOQ - også kalt [Wilson's formel](#))

Anta at forutsetningene på forrige side er oppfylt, dvs. side 211.
Den totale kostnaden for bestillings- og lagerkostnader

$$C(X) = S \frac{D}{X} + H \frac{X}{2} \quad (5.9)$$

er minimal for følgende optimal ordrestørrelse:

$$X^* = \sqrt{\frac{2SD}{H}} \quad (5.10)$$

hvor

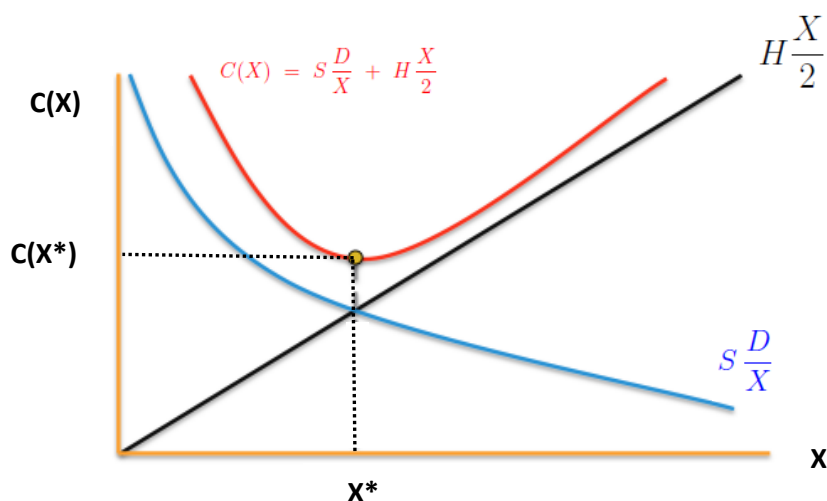
D = etterspørsel (demand) (5.11)

H = lagerkostnad per enhet (5.12)

S = bestillingskostnaden/setupkostnaden assosiert med en ny ordre (5.13)

X^* = optimal ordrestørrelse (EOQ) (5.14)

■



Figur 5.10: EOQ.

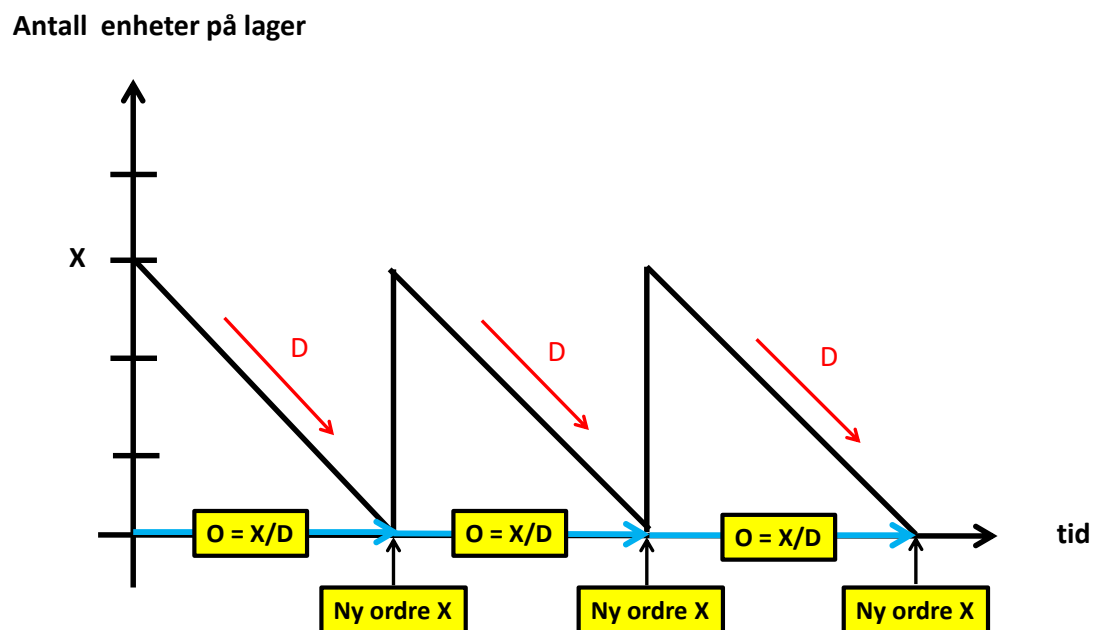
Utleddning:

Den totale kostnaden for bestillings- og lagerkostnader er:

$$C = C_{bestilling} + C_{lager} \quad (5.15)$$

Vi ønsker å **minimere totale kostnader**, dvs. summen av ordrekostnadene og lagerkostnaden.

Vi har en variabel nemlig X , som er selve ordrestørrelsen vi skal bestemme. Vi må nå først uttrykke den totale merkostnaden ved variabelen X^* . For å forstå utledningen av kostnadene, må vi se på hvordan **lagerprofilen** utarter seg, dvs grafen som viser hvordan lageret utvikler seg over tid. Figur 5.11 viser den kjente **sagtanngrafen**, som oppstår fra antakelsen om konstant etterspørsel.



Figur 5.11: Antall enheter på lager sfa. av tid.

1) Bestillingskostnadene: (setupkostnadene)

Hvis vi tenker oss at vi starter med X enheter på lager, så vet vi at lageret leverer til markedet med en konstant etterspørsel D . Siden lageret minker med D enheter pr tidsenhet, tar det nøyaktig

$$O = \frac{X}{D} \quad (\text{omløpstiden}) \quad (5.16)$$

tidsenheter for lageret å bli tomt. Denne tiden kalles **omløpstiden** til lageret. Antall bestillinger i løpet av en tidsenhet er dermed $1/O = D/X$. Hvis vi multipliserer med bestillingskostnaden, får vi totale bestillingskostander:

$$C_{\text{bestilling}} = S \frac{D}{X} \quad (5.17)$$

hvor S er bestillingskostnaden/setupkostnaden assosiert med en ny ordre.

2) Gjennomsnittlige lagerkostnader

Legg merke til at vi bruker **gjennomsnittlige lagerkostnader**. Den gjennomsnittlige lagerkostnaden kan enkelt beregnes siden vi vet at vi har X enheter når lageret er fullt, og 0 enheter når lageret er tomt. Og dette mønsteret gjentar seg hele tiden. Siden

$$H = \text{lagerkostand per enhet per tidsenhet} \quad (5.18)$$

får vi

$$C_{\text{lager}} = H \frac{X+0}{2} = H \frac{X}{2} \quad (5.19)$$

3) Totale kostnader:

Setter vi inn uttrykkene for bestillings- og lagerkostnadene, får vi at:

$$C(X) = S \frac{D}{X} + H \frac{X}{2} \quad (5.20)$$

som er en funksjon mhp. ordrestørrelsen X .

4) Optimal ordrestørrelse:

Vi kan nå finne optimum ved å finne nullpunktet til den **deriverte** av funksjonen, dvs. hvor **tangenten til grafen** til $C(X)$ er **horisontal**). Den derivert av $C(X)$ er gitt ved:

$$\frac{dC(X)}{dX} = -S \frac{D}{X^2} + \frac{H}{2} \quad (5.21)$$

Vi setter den deriverte lik null og løser likningen mhp X :

$$\frac{dC}{dX} = 0 \quad (5.22)$$

$$-S \frac{D}{X^2} + \frac{H}{2} = 0 \quad (5.23)$$

$$X^2 = \frac{2DS}{H} \quad (5.24)$$

$$X^* = \sqrt{\frac{2DS}{H}} \quad (5.25)$$

Dermed har vi funnet X^* , eller EOQ ,
dvs. den ordrestørrelsen som gir minst bestillings- og lagerkostnader.

■

Eksempel: ([Elkjøp iPhone EOQ](#))

Elkjøp har estimert en *ukentlig* etterspørselsrate på 2200 iPhoner. Vi antar at denne raten er *konstant*. De har videre estimert bestillingskostnaden til 2000 kroner pr ordre. Økonomiavdelingen ved Elkjøp opererer med kapitalrente på 0.3 i året. Innkjøpsprisen per iPhone er 2800 kroner. Anta at Apple ikke gir kvantumsrabatter og Elkjøp tillater ingen stockout.

Beregn [optimal ordrestørrelse](#) og [optimale totale kostnader](#) ved hjelp av EOQ.



Figur 5.12: Elkjøp og iPhone.

Løsning:

Fra oppgaven har vi følgende data:

$$D = 2200 \quad \text{konstant ukentlig etterspørselsrate} \quad (5.26)$$

$$S = 2000 \quad \text{bestillingskostnaden} \quad (5.27)$$

$$r = 0.3 \quad \text{kapitalrenten (årlig)} \quad (5.28)$$

$$P = 2800 \quad \text{innkjøpspris for en stk. iPhone} \quad (5.29)$$

$$H = \frac{r \cdot P}{52} = \frac{0.3 \cdot 2800}{52} = 16.15 \quad \text{ukentlig lagerkostnad pr iPhone} \quad (5.30)$$

Satt inn i Wilsons formel gir:

$$\underline{X^*} = \sqrt{\frac{2SD}{H}} \quad (5.31)$$

$$= \sqrt{\frac{2 \cdot 2200 \cdot 2000}{16.15}} = \underline{738} \quad (5.32)$$

Optimal ordrestørrelse er altså 738 iPhoner.

Optimale totale bestillings- og lagerkostnader $C(X^*)$ er dermed gitt ved:

$$\underline{C(X^*)} = S \frac{D}{X^*} + H \frac{X^*}{2} \quad (5.33)$$

$$= \left(2000 \cdot \frac{2200}{738} + 16.15 \cdot \frac{738}{2} \right) \text{NOK} = \underline{11\,923 \text{ NOK}} \quad (5.34)$$

Optimal ordrestørrelse gir en tilhørende minimal bestillings- og lagerkostnad på 11 923 NOK.



5.5 Etterbestilling - dvs. når legger vi inn nye ordrer

Vi har nå sett på EOQ som metode for å finne optimal ordrestørrelse under hovedantakselsen om at etterspørselen er gitt ved konstant rate.

Vi skal i dette avsnittet se på modeller for **når** vi legger inn en ny bestilling/produksjonsordre (etterbestilling).

Hvorfor er dette viktig?

Er det ikke bare å bestille etter hvor lang **ledetiden** er, dvs. tiden det tar å få levert/produsert varene?

Det er det **åpenbart ikke**, siden ingen etterspørsel er fullstending konstant. Det vil forekomme variasjoner, som gjør at det å bare bestille/produsere ift. ledetiden, raskt gir stockout problemer eller overlaging.

Vi må se på det **faktiske lagernivået**, og basert på dette nivået, vil vi avgjøre når vi skal legge inn en ny ordre.

Vi skiller gjerne mellom to typer etterbestillingsmodeller:

- **Bestillingspunktmodeller** - vi legger inn en ny ordre når lagernivået har nådd et bestemt nivå. Dette nivået kalles bestillingspunktet. Modellen krever at vi kjenner lagernivået til enhver tid.
- **Periodiske lagermodeller** - vi kontroller lagernivået med periodiske mellomrom, f.eks. en gang daglig. Etterbestilling bestemmes basert på observert lagernivå. (LOG500)

Vi skal kun se på bestillingspunktmodellen i dette kompendiet.

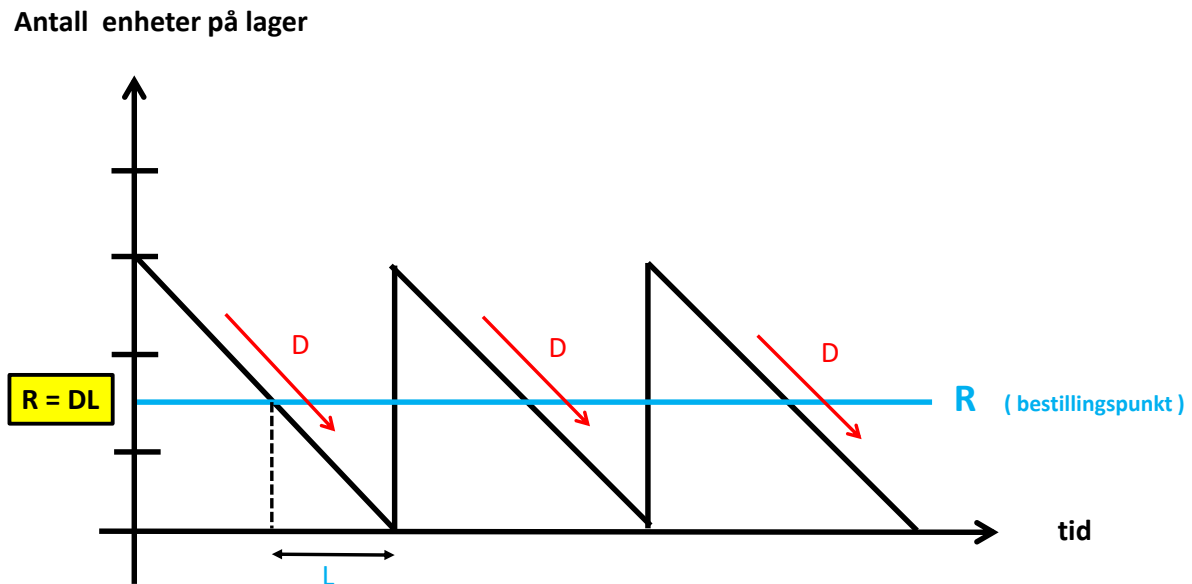
5.5.1 Bestillingspunktmodellen - uten og med sikkerhetslager

Uten sikkerhetslager

I dette avsnittet skal vi se på en bestillingspunktmodell under antakelsen om at vi har en relativ jevn etterspørselsrate D .

Vi antar i tillegg at vi har en gitt fast ordrestørrelse (f.eks. EOQ).

Vi får dermed den samme [sagtannprofilen](#) på lageret, men nå tar vi høyde for at det faktisk finnes [variasjon](#).



Figur 5.13: Antall enheter på lager sfa. av tid.

Så det store spørsmålet blir:

[Når](#) skal vi bestille?

Svaret er enkelt hvis vi ser på figur 5.13. Vi bør legge inn ordren akkurat når lagernivået er lik etterspørselen som går med under ledetiden.

Bestillingspunktet kan illustreres i figuren enkelt ved at når vi har null på lager, så går vi L tidsenheter tilbake i tid og leser av lagernivået ved dette tidspunktet. Det avleste lagernivået er det ønskede bestillingspunktet.

Vi har dermed at bestillingspunktet R , uten sikkerhetslager, er gitt ved:

$$R = DL \tag{5.35}$$

hvor D er etterspørselsraten, mens L er ledetiden.

Legg merke til at D og L , må ha samme enhet (uker, dager etc).

Setning: (bestillingspunkt R - uten sikkerhetslager)

Anta at forutsetningene for EOQ -formelen er oppfylt, se side 210.

Konstant etterspørsel og ingen stockout er noen av disse forutsetningene.

Bestillingspunktet R er da:

$$R = DL \quad (5.36)$$

hvor

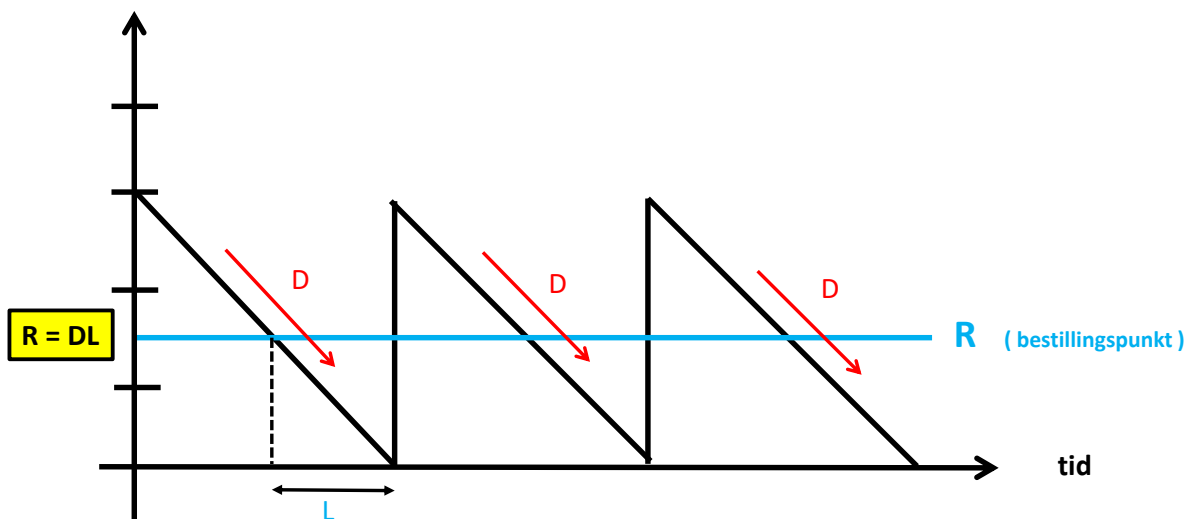
$$D = \text{etterspørsel (demand)} \quad (5.37)$$

$$L = \text{ledetid} \quad (5.38)$$

$$R = \text{bestillingspunkt} \quad (5.39)$$

■

Antall enheter på lager



Figur 5.14: Antall enheter på lager sfa. av tid.

Eksempel:

La oss igjen se på Elkjøpeksemplet hvor vi opererer med en seriestørrelse på 6000 iPhoneer (dette er *ikke* EOQ verdien). Vi har oppgitt at etterspørselsraten er på 2200 iPhoneer per uke. Anta videre at [ledetiden](#) er 14 dager.

Hva blir bestillingspunktet R ?



Figur 5.15: Elkjøp og iPhone.

Løsning:

Før vi kan beregne bestillingspunktet må vi sørge for at ledetiden og etterspørselsraten har samme tidsenhet. Siden ledetida er 14 dager, dvs. 2 uker, får vi at bestillingspunktet er gitt ved:

$$\underline{R} = DL = 2200 \cdot 2 = \underline{4400} \quad (5.40)$$

Konklusjon:

Når lagernivået har nådd 4400 iPhoner så må vi legge inn en ny ordre på 6000 iPhoner.

Legg merke til at **tidsrommet** mellom to påfølgende bestillinger **varierer**, mens ordrestørrelsen er fast.



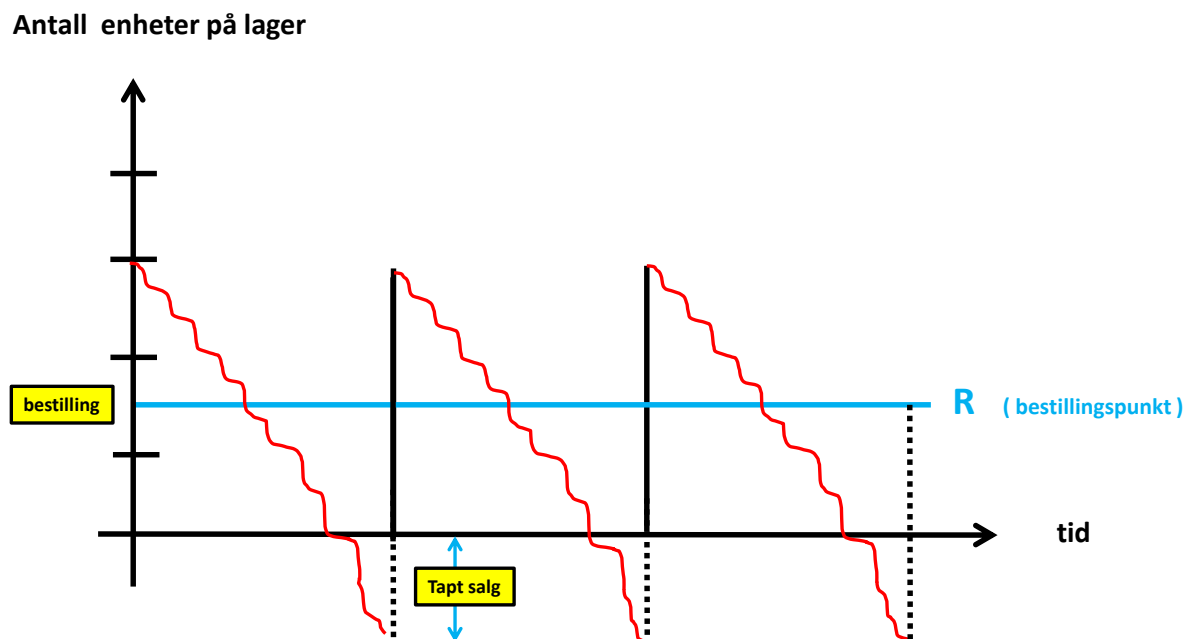
Med sikkerhetslager

Problemet med bestillingspunktmodellen som vi utledet har hittil, er at den ikke tar høyde for **variasjonene** i etterspørselen.

Anta derfor vi har følgende situasjon:

$$\text{Etterspørsel pr. tidsenhet} = D + \text{variasjon} \quad (5.41)$$

I stedet for figur 5.15 får vi nå en situasjon hvor vi faktisk kan gå tom på lager dersom vi benytter samme bestillingspunkt $R = DL$.



Figur 5.16: Usikker etterspørsel - uten sikkerhetslager.

For å ta høyde for variasjonen, må vi holde [sikkerhetslager](#). Anta vi har fastsatt et sikkerhetslager på SS enheter, da inkluderer vi ganske enkelt sikkerhetslageret:

$$R = DL + SS \tag{5.42}$$

La oss formulere dette i en egen setning.

Setning: (bestillingspunkt R - med sikkerhetslager)

Anta at forutsetningene for EOQ -formelen er oppfylt, se side 210, bortsett for at vi antar at etterspørselen har en gitt variasjon rundt D , dvs si anta likning (5.41) er tilfelle.

Bestillingspunktet R er da:

$$R = DL + SS \quad (5.43)$$

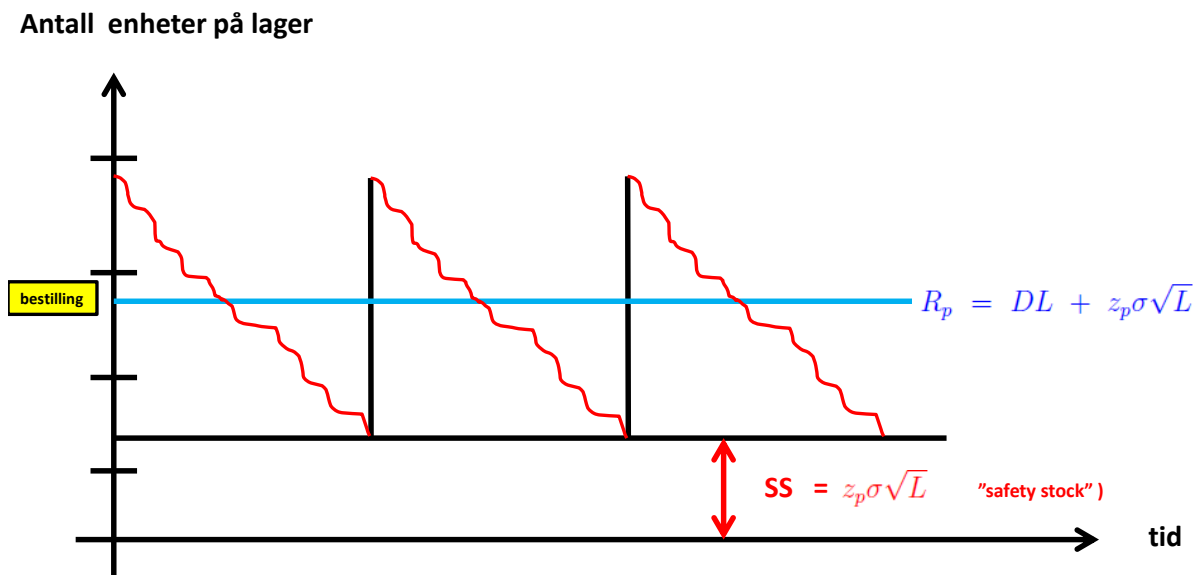
$$D = \text{etterspørsel (demand)} \quad (5.44)$$

$$L = \text{ledetid} \quad (5.45)$$

$$SS = \text{størrelsen på sikkerhetslageret} \quad (5.46)$$

$$R = \text{bestillingspunkt} \quad (5.47)$$

■



Figur 5.17: Usikker etterspørsel - med sikkerhetslager.

Så det åpenbare spørsmålet er: Hvor **stort** skal sikkerhetslageret **SS** være?

Svaret er mer nyansert enn man tror, siden det er **fullstendig avhengig av vår egen policy**. Vi må med andre ord, selv komme med hvilket kriterium som vi skal basere sikkerhetslageret på.

De to **vanligste** kriteriene er basert på statistiske mål:

- sannsynligheten for å ikke gå tom på lager (servicegraden)
- forventet antall enheter vi går tom på lager med (behandles ikke dette kompendiet)

Det første kriteriet lyder:

”Vi har fastsatt et sikkerhetslager slik at sannsynligheten for å gå tom på lager er 99 %”.

Det andre kriteriet som:

“Vi har fastsatt et sikkerhetslager slik at vi har en forventet mangel på f.eks 250 iPhoner”.

5.6 Fastsettelse av sikkerhetslager under krav om sannsynlighet for stockout

Anta vi ønsker å beregne sikkerhetslager, basert på alternativ 1 - sannsynlighet for ingen stockout.

Anta vi har valgt servicegrad på 99 %. Problemet er at for å kunne [si noe om sannsynligheter](#), må vi kjenne til den [statistiske fordelingen](#) til etterspørselen.

Den vanlige antagelsen er at etterspørselen er ”normalfordelt med forventingsverdi D og standardavvik σ ”. I praksis, betyr det at etterspørselen grovt sett vil falle innenfor intervallet $D \pm \sigma$ for 60% av dagene.

Men 60% utgjør ikke målsetningen om servicegrad på 99 % sannsynlighet for ingen *stockout*. Så spørsmålet blir dermed hvor [mange](#) standardavvik vi må øke intervallet med for å oppnå at etterspørselen faller inn under intervallet i 99% av dagene.¹

Dette antallet standardavvik kaller vi [sikkerhetsfaktoren](#) z_p , hvor p er den valgte servicegraden (99%). Denne sikkerhetsgraden finnes ved såkalt ”omvendt tabelloppslag for standard normalfordeling”. I vårt fag vil sikkerhetsfaktoren enten være oppgitt, eller så kan tallet beregnes fra oppgitt bestillingspunkt. For servicegrad $p = 0.99$ er for eksempel $z_{0.99} = 2.33$.

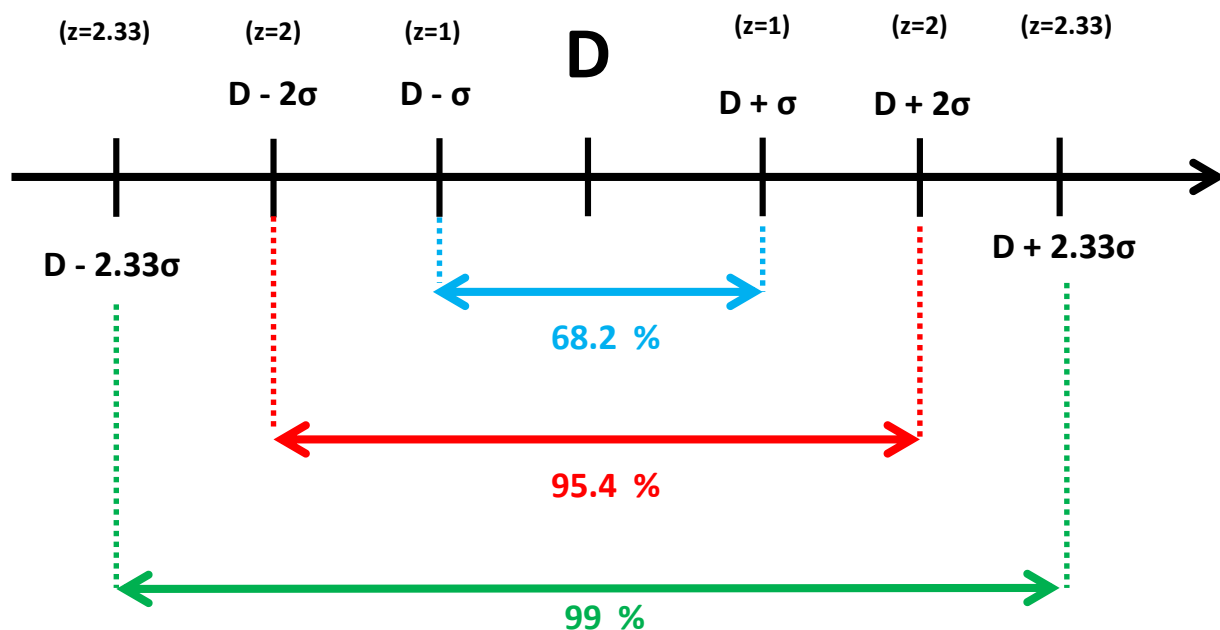
¹Dette spørsmålet krever inngående kunnskap i statistikk, og behandles nøye faget MAT110 Statistikk 1.

Vi kan under antakselsen om at etterspørselen er "normalfordelt med forventingsverdi D og standardavvik σ " og krav om servicegrad 99%, utlede at sikkerhetslageret må være lik

$$SS = z_{0.99} \sigma \sqrt{L} \quad (5.48)$$

Dersom vi velger en annen servicegrad, f.eks. 0.999, leser vi av den tilsvarende sikkerhetsfaktoren $z_{0.999} = 3.10$ fra det omvendte tabelloppslaget.

Vi kan dermed konkludere med følgende setning for fastsettelse av bestillingspunkt med sikkerhetslager:



Figur 5.18: Servicegrad p og sikkerhetsfaktor Z_p . (D = forventingsverdi av etterspørsel per tidsenhet).

Setning: (bestillingspunkt R - med sikkerhetslager)

Anta at forutsetningene for EOQ -formelen er oppfylt, se side 210, bortsett for at vi antar at etterspørselen er

normalfordelt

med forventningsverdi D og standardavvik σ .

Anta videre vi har fastsatt en servicegrad på p , dvs. $p\%$ sannsynlighet for ingen stockout.

Bestillingspunktet R_p gitt ved:

$$R_p = DL + z_p \sigma \sqrt{L} \quad (5.49)$$

hvor

$$z_p = \text{sikkerhetsfaktoren (oppgitt)} \quad (5.50)$$

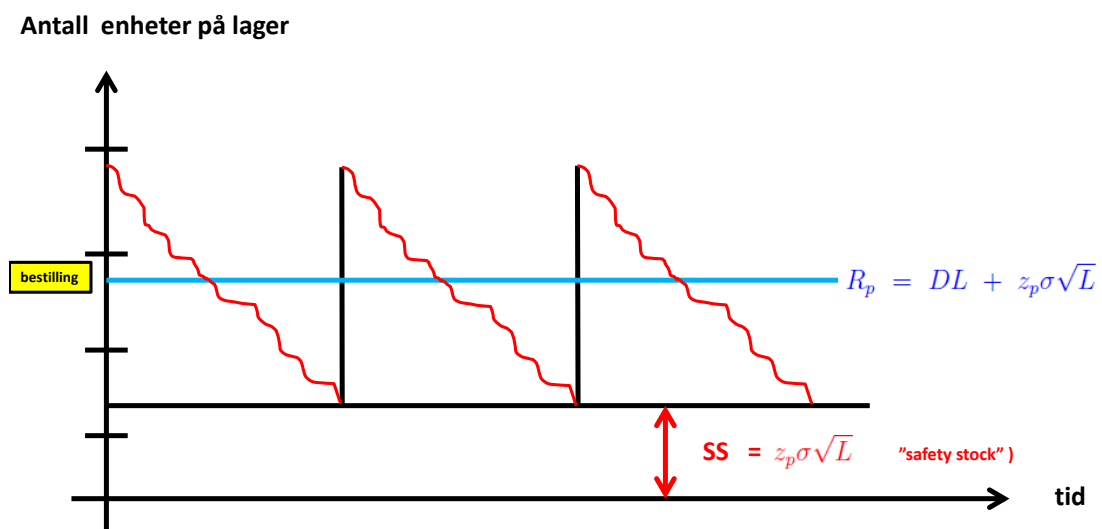
$$p = \text{servicegrad} \quad (5.51)$$

$$D = \text{forventet etterspørsel pr. tidsenhet} \quad (5.52)$$

$$\sigma = \text{standardavvik} \quad (5.53)$$

$$L = \text{ledetid} \quad (5.54)$$

■



Figur 5.19: Usikker etterspørsel - med sikkerhetslager.

Vi skal nå se på et gjennomgående eksempel som både inkluderer EOQ, bestillingspunkt og fastsettelse av sikkerhetslager under en gitt servicegrad.

Eksempel: (EOQ, bestillingspunkt og sikkerhetslager)

Brunvoll AS lager sidepropeller i verftsindustrien. Selve lakkeringsfargen på propellen er kunde-bestemt siden den som regel må matche med skroget. Brunvoll holder derfor masterplan for lakk.

Vi skal lage et bestillingssystem for den mest brukte lakken - burgunderrød, basert på EOQ og bestillingspunktmodellen. Anta vi har følgende data:

$$D = 7300 \quad (\text{antall liter pr år - normalfordelt med middelvei 7300}) \quad (5.55)$$

$$\sigma = 35 \quad (\text{ukentlig standardavvik}) \quad (5.56)$$

$$P = 290 \quad (\text{innkjøpspris per liter lakk}) \quad (5.57)$$

$$r = 0.15 \quad (\text{årlig lagerrente}) \quad (5.58)$$

$$S = 349 \quad (\text{bestillingskostnaden for lakk}) \quad (5.59)$$

$$H = rP/52 = 0.15 \cdot 290 = 43.5 \quad (\text{lagerkost per liter pr år}) \quad (5.60)$$

$$L = 2 \quad (\text{ledetid i antall dager}) \quad (5.61)$$

$$p = 0.99 \quad (\text{ønsket servicegrad}) \quad (5.62)$$

$$z_{0.99} = 2.33 \quad (\text{sikkerhetsfaktoren for gitt servicegrad}) \quad (5.63)$$

Anta også konstant etterspørsel av lakken og ingen stockout.



Figur 5.20: Brunvoll.

- a) Finn den optimale ordrestørrelsen som Brunvoll må ha for å minimalisere bestillings- og lagerkostnadene.
- b) Hva er omløpstiden på lageret? Og antall bestillinger i året?
- c) Hva er den optimalen kostnadene, dvs. hva er den minste summen av bestillings- og lagerkostnaden?
- d) Hvor stort sikkerhetslager må Brunvoll ha for å oppnå en servicegrad på 99 %? ²
- e) Hva er den årlige kostnaden på sikkerhetslageret?
- f) Hva er bestillingspunktet for lakken?

²Husk at $z_{0.99} = 2.33$ er oppgitt i oppgaven.

Løsning:

- a) Siden vi antar konstant etterspørsel og ingen stockout så gjelder EOQ-formelen:

$$\underline{X^*} = \sqrt{\frac{2DS}{H}} \quad (5.64)$$

$$= \sqrt{\frac{2 \cdot 7300 \cdot 349}{43.5}} = \underline{342} \quad (5.65)$$

Den optimale størrelsen på hver bestilling er 342 liter lakk.

- b) Omløpstiden O på lageret er:

$$\underline{O} = \frac{X^*}{D} \cdot 52 \quad (5.66)$$

$$= \frac{342}{7300} \cdot 52 = \underline{2.44} \quad (5.67)$$

Omløpstiden på lageret er 2.44 uker.

Antall bestillinger i året blir dermed:

$$\underline{\underline{\text{antall bestillinger i året}}} = \frac{D}{X^*} \quad (5.68)$$

$$= \frac{7300}{342} = \underline{\underline{21.3}} \quad (5.69)$$

- c) De optimale merkostnadene er:

$$\underline{\underline{C^*}} = C(X^*) \quad (5.70)$$

$$= \frac{DS}{X^*} + \frac{X^*H}{2} = \left(\frac{7300 \cdot 349}{342} + \frac{342 \cdot 43.5}{2} \right) \text{ NOK/året} \quad (5.71)$$

$$= \underline{\underline{14\,888 \text{ NOK/år}}} \quad (5.72)$$

- d) Ledetiden er $2/7 = 0.2357$ dager. Dermed kan vi regne ut sikkerhetslageret dersom vi ønsker en servicegrad på 99 %:

$$\underline{\underline{SS}} = Z_{0.99} \sqrt{L/7} \sigma \quad (5.73)$$

$$= 2.33 \cdot \sqrt{2/7} \cdot 35 = \underline{\underline{44}} \quad (5.74)$$

Det vil si, vi må holde et sikkerhetslager på 44 liter lakk.

- e) Den årlige kostnaden på sikkerhetslageret er da gitt ved:

$$\underline{\underline{C_{SS}}} = H \cdot SS \quad (5.75)$$

$$= 43.5 \cdot 44 \text{ NOK/år} = \underline{\underline{1914 \text{ NOK/år}}} \quad (5.76)$$

e) Bestillingspunktet $R_{0.99}$ er da:

$$\underline{R_{0.99}} = \frac{DL}{365} + SS \quad (5.77)$$

$$= \frac{7300 \cdot 2}{365} + 44 = \underline{84} \quad (5.78)$$

Når lageret har nådd 84 liter legges inn en ny bestilling på 342 liter.

■

Vi er nå ferdig med tilfeller hvor vi antar konstant etterspørsel.

I neste avsnitt går nå over til tilfellet hvor etterspørselen er for **varierende** til at at EOQ kan benyttes. Det innebærer at vi må igjen benytte matematisk modellering.

5.7 Varierende etterspørsel - Wagner-Within

Vi har hittil antatt at [etterspørselen](#) har vært [konstant](#), men i de fleste tilfeller vil det alltid forekomme variasjoner som vi kunne håndtere ved hjelp av sikkerhetslager. Men hvis etterspørselen i seg selv er [varierende](#), som f.eks. ved [sesongprodukter](#), vil antakelsen om konstant etterspørsel bli brutt og vi må lage nye modeller for denne situasjonen.

Vi skal i dette kapitlet se på en svært fundamental modell for masterplanlegging med varierende etterspørsel. Metoden bestemmer en masterplan for en gitt horisont, med mål om å minimere totale bestillings- og lagerkostnader.³

- [Uten produksjonskapasitetsføring](#) - Wagner-Within

Denne modellen er svært fundamental og er utgangspunktet for mer kompliserte modeller.



Figur 5.21: Harvey M. Wagner og Thomson M. Within.

³Bestillingskostnader kalles også omstillingskostnader.

5.7.1 Matematisk modell - uten kapasitetsføringer

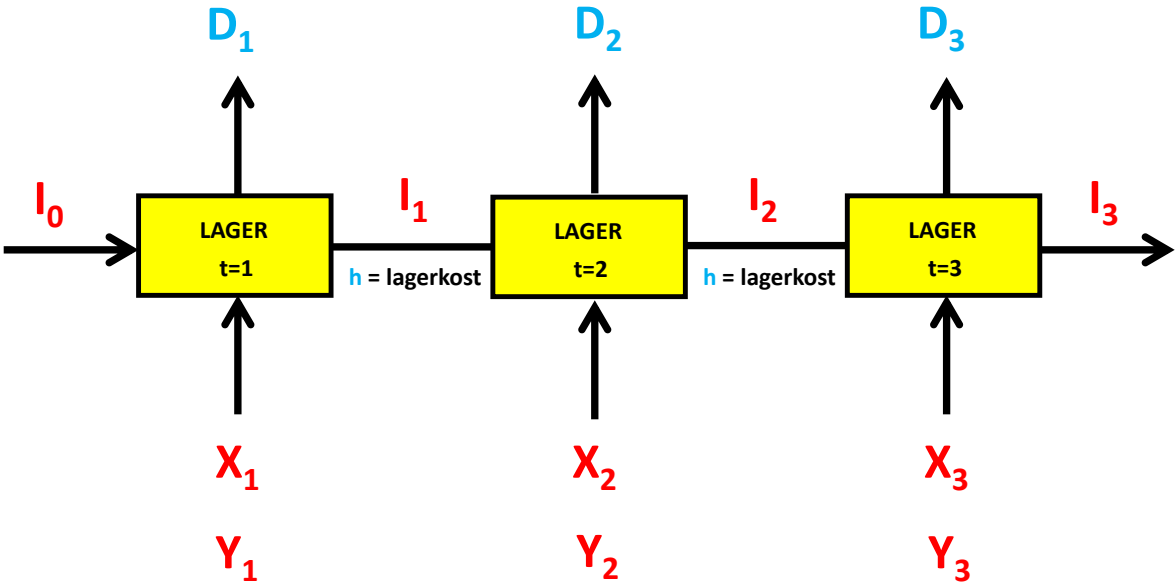
Vi skal nå se på kanskje den mest fundamentale matematiske modellen for masterplanlegging:

- skal bestemme masterplanen for et produkt, over en horisont på T uker.
- for hver periode har en gitt etterspørsel (kan være varierende).
- all etterspørsel skal oppfylles, dvs. ingen stockout.
- ikke finnes noen kapasitetsføringer på bestillingsmengde/produksjonsmengde.
- gitt lagerkostnad over hele horisonten (per vareenhet).
- fast bestillingskostnad/setupkostnad.
- gitt startslager og sluttlager.

Vi starter med å formulere det matematiske problemet, hvor vi søker å minimere totale kostnader. Som alltid splitter vi opp modelleringen i

- 1) data
- 2) variabler
- 3) målfunksjon
- 4) føringer

Illustrasjon av eksempel med periode $T = 3$:



Figur 5.22: Illustrasjon av dataene D_t og h samt variablene X_t , I_t og Y_t for $T = 3$.

1) Data:

Siden denne modellen er [generell](#), må vi benytte indeksnotasjon. Vi har kun en indeks for dette problemet, nemlig periodene indekseres ved $t = 1, 2, \dots, T$.

$$I_0 = \text{start/initielt lager inn til uke 1} \quad (5.79)$$

$$I_T = \text{sluttlager ved utgangen av periode } T \quad (5.80)$$

$$D_t = \text{etterspørsel for uke } t, \text{ hvor } t = 1, 2, \dots, T \quad (5.81)$$

$$h = \text{overlagringskostnad per enhet i en periode} \quad (5.82)$$

$$S = \text{bestillingskostnad/omstillingskostnad per batch/serie} \quad (5.83)$$

2) Variabler:

Vi har tre typer variabler i problemet:

- a) **ordrestørrelse** - hvor mange enheter vi skal bestille/produsere pr periode
- b) **lagernivå** - hvor mange enheter vi skal ha på lager fra en periode til neste
- c) **bestille/ikke bestille** - ja/nei variabel som avgjør om vi skal ha bestilling/produksjon eller ikke

a) **Ordrestørrelser/seriestørrelser:**

$$X_t = \text{antall enheter bestilt/produsert til periode } t \quad (5.84)$$

hvor $t = 1, 2, \dots, T$.

b) **Lagernivå:**

$$I_t = \text{antall enheter overlagret fra periode } t \text{ til periode } t + 1 \quad (5.85)$$

hvor $t = 1, 2, \dots, T - 1$.

c) **Bestille/ikke bestille** (samme for produksjon):

$$Y_t = \begin{cases} 1 & , \text{ dersom vi bestiller/produserer i periode } t \\ 0 & , \text{ hvis ikke} \end{cases} \quad (5.86)$$

hvor $t = 1, 2, \dots, T$.

3) Målfunksjon:

Totale kostnader fås ved å summere bestillingskostnadene/setupkostnadene og lagerkostnadene for alle periodene i horisonten:

$$C = \sum_{t=1}^T S Y_t + \sum_{t=1}^{T-1} h I_t \quad (5.87)$$

4) Føringer:

Vi har to føringer i modellen:

- a) **lagerbalansen** - som sørger for at all etterspørsel oppfylles
- b) **logisk sammenheng** mellom ordrestørrelse (X_t -variablene) og hvorvidt vi bestiller eller ikke (Y_t -variablene)

a) **Lagerbalansen:**

Husk at lagerbalansen er lik innkommende lager pluss lager in minus lager ut lik utgående lager:

$$I_{t-1} + X_t - D_t = I_t \quad (5.88)$$

som er lagerbalanse for periode t , hvor $t = 1, 2, \dots, T$.

b) **Logisk sammenheng:**

Vi kan uttrykke føringen *logisk* ved:

$$\text{Hvis } Y_t = 0 \text{ så må } X_t = 0 \quad (5.89)$$

for hver periode t , hvor $t = 1, 2, \dots, T$.

Alternativt kan vi skrive denne føringen på matematisk ikke-linær form:

$$X_t(1 - Y_t) = 0 \quad (5.90)$$

for hver periode t , hvor $t = 1, 2, \dots, T$, eller på linær form:

$$X_t \leq MY_t \quad (5.91)$$

for hver periode t , hvor $t = 1, 2, \dots, T$, hvor $M = \sum_t^T D_t$ er summen av all etterspørsel i horisonten.

■

Eksempel: ([Glamox masterplan Wagner-Withins modell - 8 uker](#))

Anta at et av halvfabrikatene hos Glamox har prognoser for de neste 8 ukene med varierende etterspørsel:

Uke t	1	2	3	4	5	6	7	8
Etterspørsel D_t	300	400	100	200	500	100	150	400

Tabell 5.3: Ettersørsel D_t .

Glamox har estimert en lagerkostnad på 10 kroner pr lysarmatur pr uke, mens bestillingskostnaden er på 3100 kroner. Anta også at de ønsker å legge opp et sikkerhetslager i slutten av uke 8 på 100 lysarmaturer i løpet av perioden. initielt lager i uke 1 er 50 lysarmaturer.

Skriv opp matematisk modell for minimering av totale kostnader, dvs. totale lager- og omstillingskostnader.⁴

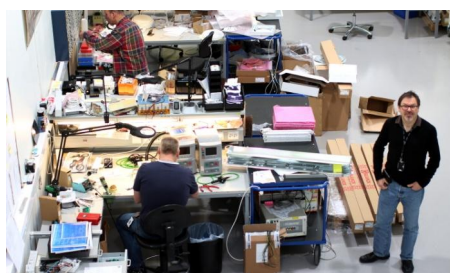


Foto: Arild J. Waagbø

Figur 5.23: Glamox.

⁴Vi har allerede utledet den generelle Wagner-Within modellen så det eneste vi trenger å gjøre er å legge inn de oppgitte datastørrelsene.

Løsning:

1) Data:

Vi har følgende oppgitte data:

$$I_0 = 50 \quad \text{start/initielt lager inn til uke 1} \quad (5.92)$$

$$I_8 = 100 \quad \text{sluttlager ved utgangen av periode } T \quad (5.93)$$

$$(5.94)$$

$$D_t = \text{etterspørsel for uke } t, \text{ hvor } t = 1, 2, \dots, 8 \quad (5.95)$$

$$h = 10 \quad \text{overlagringskostnad per enhet i en periode} \quad (5.96)$$

$$S = 3100 \quad \text{bestillingskostnad/omstillingskostnad per batch/serie} \quad (5.97)$$

2) Variabler:

a) Ordrestørrelser/seriestørrelser:

$$X_t = \text{antall enheter bestilt/produsert til periode } t \quad (5.98)$$

hvor $t = 1, 2, \dots, 8$.

b) Lagernivå:

$$I_t = \text{antall enheter overlagret fra periode } t \text{ til periode } t + 1 \quad (5.99)$$

hvor $t = 1, 2, \dots, 7$.

c) Bestille/ikke bestille (samme for produksjon):

$$Y_t = \begin{cases} 1 & , \text{ dersom vi bestiller/produserer i periode } t \\ 0 & , \text{ hvis ikke} \end{cases} \quad (5.100)$$

hvor $t = 1, 2, \dots, 8$.

3) Målfunksjon:

Totale kostnader fås ved å summere bestillingskostnadene/setupkostnadene og lagerkostnadene for alle periodene i horisonten:

$$C = \sum_{t=1}^8 S Y_t + \sum_{t=1}^7 h I_t \quad (5.101)$$

4) Føringer:

Vi har to føringer i modellen:

- a) **lagerbalansen** - som sørger for at all etterspørsel oppfylles
- b) **logisk sammenheng** mellom ordrestørrelse (X-variablene) og hvorvidt vi bestiller eller ikke (Y-variablene)

a) **Lagerbalansen:**

Husk at lagerbalansen er lik innkommende lager pluss lager in minus lager ut lik utgående lager:

$$I_{t-1} + X_t - D_t = I_t \quad (5.102)$$

som er lagerbalanse for periode t , hvor $t = 1, 2, \dots, T$.

b) Logisk sammenheng:

Vi kan uttrykke føringen *logisk* ved:

$$\text{Hvis } Y_t = 0 \text{ så må } X_t = 0 \quad (5.103)$$

for hver periode t , hvor $t = 1, 2, \dots, 8$.

Alternativt kan vi skrive denne føringen på matematisk ikke-linær form:

$$X_t(1 - Y_t) = 0 \quad (5.104)$$

for hver periode t , hvor $t = 1, 2, \dots, 8$, eller på linær form:

$$X_t \leq MY_t \quad (5.105)$$

for hver periode t , hvor $t = 1, 2, \dots, 8$, hvor $M = \sum_t = 1^T D_t$ er summen av all etterspørsel i horisonten.

■

$$\begin{aligned} \text{Min} \quad & 3100(Y1 + Y2 + Y3 + Y4 + Y5 + Y6 + Y7 + Y8); \\ & + 10(I1 + I2 + I3 + I4 + I5 + I6 + I7); \end{aligned}$$

$$\begin{aligned} 50 + X1 - 300 &= I1; \\ I1 + X2 - 400 &= I2; \\ I2 + X3 - 100 &= I3; \\ I3 + X4 - 200 &= I4; \\ I4 + X5 - 500 &= I5; \\ I5 + X6 - 100 &= I6; \\ I6 + X7 - 150 &= I7; \\ I7 + X8 - 400 &= 100; \end{aligned}$$

$$\begin{aligned} X1 &\leq 2150 * Y1; & \text{!Tallet 2150 er lik summen av all etterspørsel;} \\ X2 &\leq 2150 * Y2; \\ X3 &\leq 2150 * Y3; \\ X4 &\leq 2150 * Y4; \\ X5 &\leq 2150 * Y5; \\ X6 &\leq 2150 * Y6; \\ X7 &\leq 2150 * Y7; \end{aligned}$$

$$\begin{aligned} @bin(Y1); \\ @bin(Y2); \\ @bin(Y3); \\ @bin(Y4); \\ @bin(Y5); \\ @bin(Y6); \\ @bin(Y7); \\ @bin(Y8); \end{aligned}$$

Vi har nå beskrevet modellen i LINGO. Neste steg er å se på *løsningsen* LINGO gir oss.

$$C^* = C_{11011001} = 20500 \quad (\text{Optimal kostnad})$$

$$X1 = 250$$

$$X2 = 500$$

$$X3 = 0$$

$$X4 = 200$$

$$X5 = 750$$

$$X6 = 0$$

$$X7 = 0$$

$$X8 = 500$$

$$I1 = 0$$

$$I2 = 100$$

$$I3 = 0$$

$$I4 = 0$$

$$I5 = 250$$

$$I6 = 150$$

$$I7 = 0$$

$$Y1 = 1$$

$$Y1 = 1$$

$$Y1 = 0$$

$$Y1 = 1$$

$$Y1 = 1$$

$$Y1 = 0$$

$$Y1 = 0$$

$$Y1 = 1$$

Kommentarer:

1. $C_{11011001}$ er kostnaden vi får gitt at $Y_1 = 1$, $Y_2 = 1$, $Y_3 = 0$, $Y_4 = 1$, $Y_5 = 1$, $Y_6 = 0$, $Y_7 = 0$ og $Y_8 = 1$. Denne notasjonen blir viktig i neste kapittel.
2. Legg merke til at alle produksjonsvariablene (X -ene) er en sum av [hele etterspørselsbehov](#). Denne egenskapen kalles [dominante produksjonsplaner](#), og blir også viktig i neste kapittel.

5.7.2 Wagner-Within algoritmen

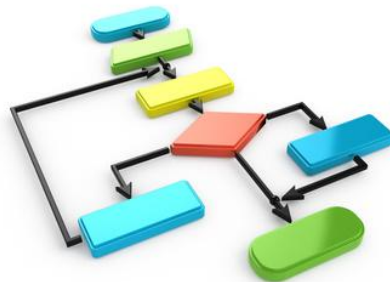
I forrige avsnitt beskrev vi den fundamentale masterplanmodellen når vi ikke har kapasitetsføringer. På grunn av binærvariablene (ja/nei), så vokser dette problemet eksponensielt raskt med antall perioder.

Vi har totalt T^2 kombinasjoner for hvilke perioder vi skal bestille/ikke bestille. Lingo klarer fint å løse problemer av denne typen når $T = 8$ som var tilfelle i vårt eksempel fra forrige kapittel. Men hva skjer når vi øker antall produkter og antall perioder?

Lingo feiler!

Det viser seg at problemet har mye struktur i seg som kan benyttes for å løse denne masterplanen mye mer effektivt.

Wagner-Within resonnerte seg frem til 4 observasjoner som ledet til en mye mer effektiv algoritme for å løse dette problemet. Algoritmen kalles Wagner-Within algoritmen og er en unik måte å vise hvordan fornuftige planleggingsprinsipper kan overføres til effektive løsningsalgoritmer.



Figur 5.24: Wagner-Within algoritmen.

De fire observasjonene til Wagner-Within:

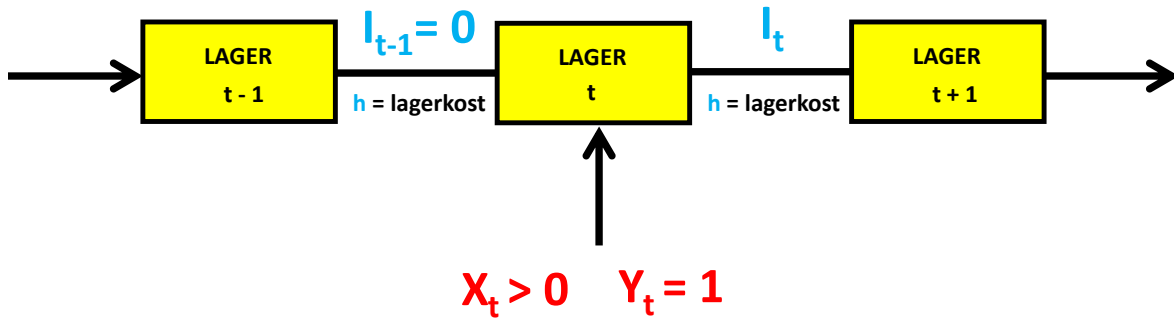
1. **Ingen overlagring til en periode med bestilling.**
2. **Dominante produksjonsplaner** - det er optimalt å kun bestille *hele* produksjonsbehov.
3. **Splitting i subproblemer** - hvis det er optimalt at det er ingen overlagring fra en periode t , så kan vi splitte problemet i to intervaller, $[1, t - 1]$, og $[t, T]$.
4. **Horisontteoremet** - dersom det for et gitt subproblem $[1, \hat{t}]$ er optimalt at **siste bestilling** skjer i en tidligere periode $\hat{\hat{t}} < \hat{t}$, så kan vi løse subproblemene $[1, \hat{\hat{t}} - 1]$ for seg selv (dvs vi kan kutte disse periodene vekk ifra algoritmen, da vi har allerede regnet ut dette subproblemene i algoritmen). Hvis siste bestilling skjer i periode \hat{t} , dvs $\hat{\hat{t}} = \hat{t}$, så er det optimalt å produsere i periode \hat{t} .

Vi skal nå gå gjennom logikken bak hver av observasjonene i listen over. Vi er klar over at spesielt punkt 4 kan virke abstrakt.

Observasjon 1

(ingen overlagring til en periode med bestilling)

Observasjon 1: dersom $X_t > 0$ så er det optimalt at $I_{t-1} = 0$.



Figur 5.25: Dersom det er produksjon i periode t , dvs. $Y_t = 1$, så er det optimalt at $I_{t-1} = 0$.

Logikken er som følger:

Anta vi har produksjon i en periode t , dvs. anta at:

$$Y_t = 1 \quad \text{og} \quad X_t > 0 \quad (5.106)$$

Dersom vi har overlagring til periode t , dvs. $I_{t-1} > 0$, så medfører denne overlagringen medfører en lagerkostnad:

$$\text{lagerkostnad} = hI_{t-1} \quad (5.107)$$

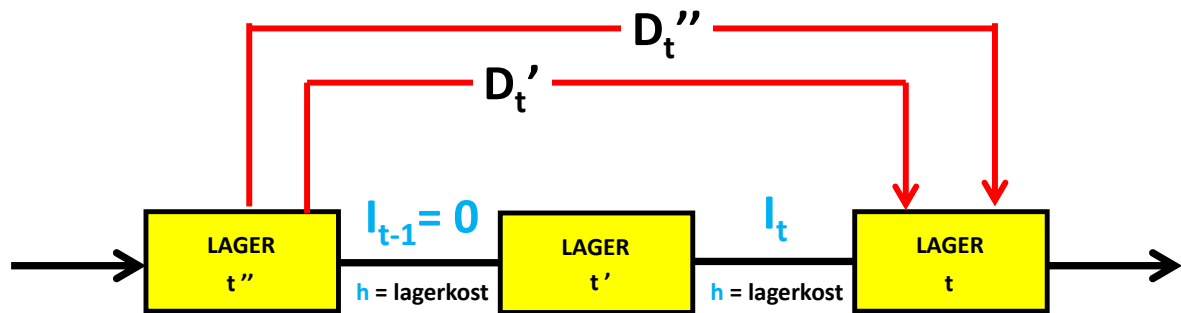
Siden vi allerede har bestilling/produksjon i periode t , så påløper det ingen ekstra kostnader dersom vi flytter lageret til periode t . Derimot sparer vi selve lagerkostnaden siden vi produserer i samme periode som vi trenger varene. Altså det er optimalt med:

optimalt: $Y_t = 1 \Rightarrow I_{t-1} = 0$ (5.108)

Observasjon 2

(dominante produksjonsplaner)

Observasjon 2: år vi først bestiller (produserer), så bestiller vi kun en sum av **hele etterspørselsbehov** - vi splitter med andre ord aldri opp etterspørselen for en gitt periode i to eller flere bestillinger.



Figur 5.26: Å splitte etterspørselen for en gitt periode er ikke optimalt.

Logikken er som følger:

Anta vi har splittet opp etterspørselen D_t for en gitt periode t i to deler: $D_{t'}$ og $D_{t''}$. Da er følgelig $D_t = D_{t'} + D_{t''}$. Anta nå at $t'' < t'$, som vist i figuren over. Siden vi allerede har lager fra t' kan vi nå flytte lageret fra t'' til t' uten at det løper ekstra kostnader. Derimot sparer vi selve lagerkostnaden mellom t'' og t' , dvs. vi

$$\text{sparer } hD_{t''}(t' - t'') \quad (5.109)$$

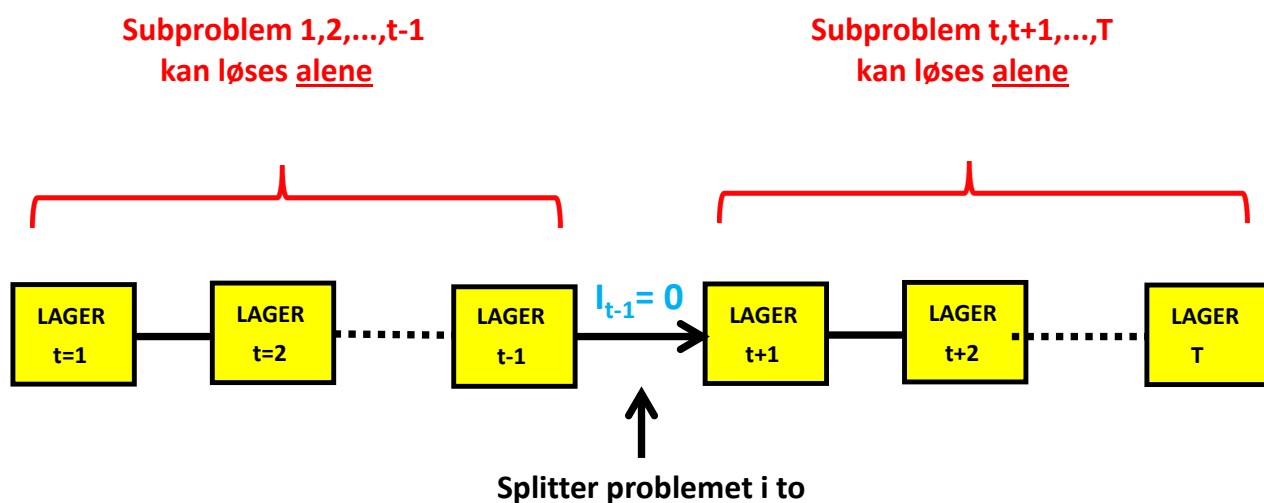
Vi har med andre ord fått en billigere plan ved å slå sammen ordrene i en ordre som dekker all etterspørsel.

Observasjon 3

(splitting i subproblemer)

Observasjon 3: det er optimalt å ha produksjon i en gitt periode t , så kan vi splitte horisonten i to delproblemer med intervallene: $[1, \dots, t-1]$ og $[t, \dots, T]$.

Legg merke til at dette er et såkalt nødvendig optimumskrav. Vi kjenner jo ikke optimum, men hvis vi likevel visste at det var optimalt å produsere i en periode t , så kan vi gjøre denne splittingen.



Figur 5.27: Hvis det er optimalt å produsere i en periode t , så kan problemet splittes i **to subproblemer**.

Logikken er som følger:

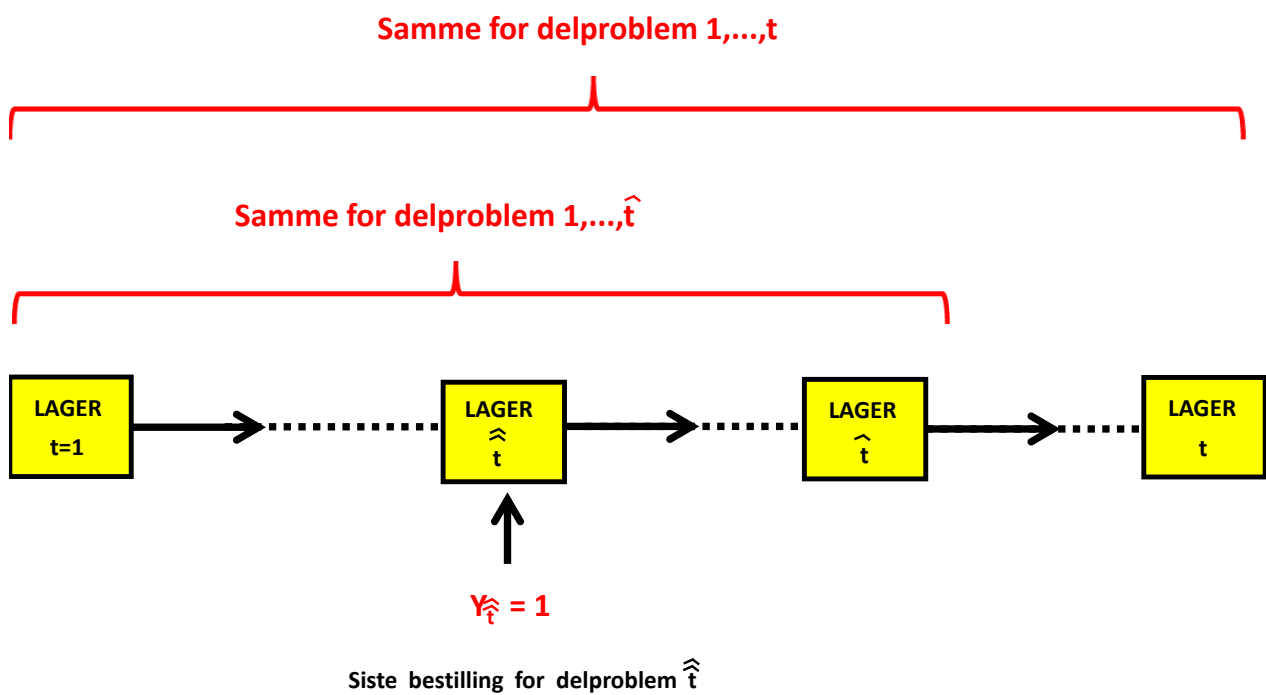
Hvis vi har produksjon i periode t , dvs. $X \geq 0$, så vet vi fra observasjon 1 at det er ingen overlagring til periode t , dvs. $I_{t-1} = 0$. Legg merke til at det er kun balanselikningene som kobler periodene sammen siden sluttlageret i den ene perioden er startslageret i neste. Men hvis sluttlageret er null, som i dette tilfellet, er også startslageret null i neste periode, noe som betyr at vi starter helt på nytt i denne perioden. Altså:

vi kan splitte horisonten i to delproblemer (5.110)

Observasjon 4

(Horisontteoremet)

Dette er det siste, men også det aller viktigste teoremet. Det er dette teoremet som faktisk reduserer antall kombinasjoner vi må regne på for å finne optimum.



Figur 5.28: Horisontteoremet.

For den optimale løsningen gjelder:

Anta vi står i en periode \hat{t} og ser tilbake på foregående periode hvor vi har bestilling, si $\hat{\hat{t}}$.

Hvis $\hat{\hat{t}} = \hat{t}$, kan vi ved observasjon 3 glemme hva som har skjedd tidligere, siden vi kan splitte problemet i to deler.

Hvis derimot $\hat{\hat{t}} < \hat{t}$, vet vi at det er optimalt at det er bestilling i periode $\hat{\hat{t}}$, men igjen fra observasjon 3 kan vi da se bort ifra hva som har skjedd før $\hat{\hat{t}}$, siden problemet kan splittes i to deler fra denne perioden.

Eksempel: ([Wagner-Within algoritmen - Glamox fortsettelse](#))

Nå skal vi forstette med eksemplet om Glamox på fra side 246.

På sidene 247-251 formulerte vi den matematiske modellen for problemet.

På sidene 252-253 løste vi problemet ved hjelp av dataprogrammet LINGO.

Nå skal vi løse modellen “*manuelt*” - ved hjelp av Wagner-Withins (WW) algoritme.

WW algoritmen er basert på de fire observasjonene nevnt på side 255.

La oss skrive opp tabellen fra side 246 en gang til:

Uke t	1	2	3	4	5	6	7	8
Etterspørsel D_t	300	400	100	200	500	100	150	400

Tabell 5.4: Ettersørsel D_t .

Fra side 247 vet vi at Glamox har følgende data:

$$I_0 = 50 \quad \text{start/initielt lager inn til uke 1} \quad (5.111)$$

$$I_8 = 100 \quad \text{sluttlager ved utgangen av periode } T \quad (5.112)$$

$$h = 10 \quad \text{overlagringskostnad per enhet i en periode} \quad (5.113)$$

$$S = 3100 \quad \text{bestillingskostnad/omstillingskostnad per batch/serie} \quad (5.114)$$

Finn variablene Y_t , X_t , I_t og minimum kostnadfunksjonen C^* .

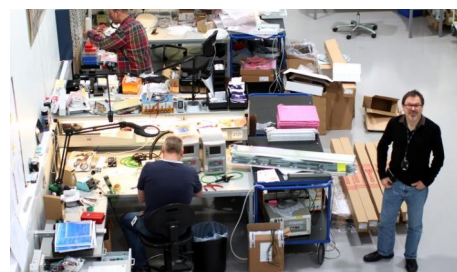


Foto: Arild J. Waagbø

Figur 5.29: Glamox.

Løsning:

Wagner-Withins algoritme er en metode som løser problemet ved å løse

subproblemer

hvor vi legger til en ny periode for hvert steg. Vi starter med første periode og deretter løser vi neste subproblem med to perioder og slik fortsetter vi inntil vi har løst problemet med alle periodene.

Algoritme

1. Løs problemet for kun **første periode**.
2. Løs (del)problemene for **neste periode**:
 - (a) Vi betinger når **siste** bestilling skjer, og regner ut kostnaden for hvert av disse tilfellene. Disse kostnadene regnes ut effektivt ved å bruke kostnadene allerede utregnet fra tidligere steg hvor adderer **de nye kostnadene som påløper** pga. den nye perioden som ble lagt til.
 - (b) Huk av det billigste alternativet.
 - (c) Benytt horisontteoremet til å kutte de periodene **før** siste bestilling (for det billigste alternativet).
3. Har vi flere perioder? Hvis ja, gå til steg 2, hvis ikke, gå til steg 4.
4. Vi har ikke **flere perioder** igjen, og har dermed funnet minimal kostnad C^* . Vi finner optimal **plan** ved:
 - (a) Start i **siste** periode.
 - (b) Finn den perioden hvor siste **bestilling** er optimal (som ble huket av i steg 2.(b)), og sett tilsvarende Y -variabel lik 1. Fortsett denne prosessen inntil vi har nådd første periode. Vi har nå funnet de optimale bestillingsperiodene (og dermed også optimal plan).
 - (c) Bruk observasjonen om **dominante produksjonsplaner** og les av tilsvarende X_t -verdier (antall produserte enheter) og I_t -verdier (antall overlagrede enheter) som er sum av hele etterspørselsbehov.

Kommentarer:

- For hvert delproblem, betinger vi på når siste bestilling skjer, vi **regner ut kostnaden** for hvert av disse tilfellene, og den med lavest kostnad gir optimum for delproblemet.
- Når vi har funnet optimum for et delproblem, sjekker vi om hvorvidt vi kan benytte **Horisontteoremet** eller ikke. Hvis vi kan benytte Horisontteoremet, kan vi fjerne et tidligere delproblem for de resterende delproblemene, og på den måten blir algoritmen mye mer effektiv.
- Algoritmen regner først ut de minimale kostnadene

$$C_t^* = \text{minimale kostnadene} \quad (5.115)$$

for hvert subproblem t , hvor $t = 1, 2, 3, \dots, 8$.

Deretter finner vi optimal plan ved å starte bakfra og finne hvor siste bestilling skjer. For å kunne gjøre dette effektivt, understreker vi for hvert subproblem, hvor siste bestilling er optimal.

- Det er hensiktsmessig å definere følgende størrelse: (for periode 4)⁵
 C_{xxxxxx} , hvor hver x er enten 0 eller 1 og representerer de fire Y -verdiene i kronologisk rekkefølge.

$$C_{1000} = \text{optimal kostnad for når siste bestilling skjer i 1. periode} \quad (5.116)$$

$$C_{x100} = \text{optimal kostnad for når siste bestilling skjer i 2. periode} \quad (5.117)$$

$$C_{xx10} = \text{optimal kostnad for når siste bestilling skjer i 3. periode} \quad (5.118)$$

$$C_{xxx1} = \text{optimal kostnad for når siste bestilling skjer i 4. periode} \quad (5.119)$$

Altså i C_{xxxx} er hver x enten 0 eller 1 og representerer de fire Y_t -verdiene i kronologisk rekkefølge, hvor **1= bestilling** i periode t og **0= ikke bestilling** i periode t .

Vi skal nå følge algoritmen og løse problemet.

For hvert steg, refererer vi til hvilket steg vi er i algoritmen beskrevet over.

⁵Tilsvarende for de andre periodene.

Steg 1:

Periode 1: (subproblem med kun første periode)

Med en eneste periode må vi ha produksjon siden vi må oppfylle etterspørselen i periode 1. Vi har altså følgende optimale løsning for dette subproblemet:

$$C_1^* = S = 3100 \quad (5.120)$$

Legg merke til at det er alltid optimalt å bestille i første periode, med mindre startslageret er tilstrekkelig til å dekke all etterspørsel i første periode.⁶

⁶Her er $I_0 = 50$ mens $D_1 = 300$ så startlageret er for lite.

Steg 2 og 3 (repetert)

Periode 2: (subproblem de 2 første periodene)

Vi betinger på siste bestilling skjer i enten 1 eller andre periode: ⁷

$$\underline{C_2^*} = \min \left[C_{10}^* , C_{x1}^* \right] \quad (5.121)$$

$$= \min \left[(siste\ bestilling\ i\ 1.\ perioden) , (siste\ bestilling\ i\ 2.\ perioden) \right] \quad (5.122)$$

$$= \min \left[(C_1^* + h \cdot D_2) , (C_1^* + S) \right] \quad (5.123)$$

$$= \min \left[3100 + 10 \cdot 400 , 3100 + 3100 \right] \quad (5.124)$$

$$= \min \left[7100 , \underline{6200} \right] \quad (5.125)$$

$$= \underline{6200} \quad (5.126)$$

Vi har streket under for hvilken periode det er optimalt å ha siste bestilling, som i dette tilfellet er periode 2.

Siden det er optimalt å bestille i periode 2 kan vi via Horisontteoremet kutte periode 1 fra videre beregninger.

⁷Siden er det 2 perioder så har C_{xx} to indekser i lign.(5.121).

Periode 3: (subproblem de 3 første periodene)

Vi starter dermed beregningene fra periode 2: ⁸

$$\underline{C}_3^* = \min \left[C_{x10}^* , C_{xx1}^* \right] \quad (5.127)$$

$$= \min \left[(siste\ bestilling\ i\ 2.\ perioden) , (siste\ bestilling\ i\ 3.\ perioden) \right] \quad (5.128)$$

$$= \min \left[(C_{x1}^* + h \cdot D_3) , (C_2^* + S) \right] \quad (5.129)$$

$$= \min \left[(6200 + 10 \cdot 100) , (6200 + 3100) \right] \quad (5.130)$$

$$= \min \left[\underline{7200} , 9300 \right] \quad (5.131)$$

$$= \underline{7200} \quad (5.132)$$

Vi har streket under for hvilken periode det er optimalt å ha siste bestilling, som i dette tilfellet også er periode 2.

Vi kan dermed via Horisontteoremet kutte periode 1 fra videre beregninger (som vi allerede har gjort).

⁸Siden er det 3 perioder så har C_{xxx} tre indekser i lign.(5.127).

Periode 4: (subproblem de 4 første periodene)

Vi starter dermed beregningene fra periode 2: ⁹

$$\underline{C}_4^* = \min \left[C_{x100}^*, C_{xx10}^*, C_{xxx1}^* \right] \quad (5.133)$$

$$= \min \left[(siste\ best.\ i\ 2.\ periode), (siste\ best.\ i\ 3.\ periode), (siste\ best.\ i\ 4.\ periode) \right]$$

$$= \min \left[(C_{x10}^* + 2 \cdot h \cdot D_4), (C_{xx1}^* + h \cdot D_4), (C_3^* + S) \right] \quad (5.134)$$

$$= \min \left[(7200 + 2 \cdot 10 \cdot 200), (9300 + 10 \cdot 200), (7200 + 3100) \right] \quad (5.135)$$

$$= \min \left[11\ 200, 11\ 300, \underline{10\ 300} \right] \quad (5.136)$$

$$= \underline{10\ 300} \quad (5.137)$$

Vi har streket under for hvilken periode det er optimalt å ha siste bestilling, som i dette tilfellet er periode 4.

Vi kan dermed via Horisontteoremet kutte fra periode 3 mhp. videre beregninger.

⁹Siden er det 4 perioder så har C_{xxxx} fire indekser i lign.(5.133).

Periode 5: (subproblem de 5 første periodene)

Vi starter dermed beregningene fra periode 3: ¹⁰

$$\underline{C_5^*} = \min \left[C_{xxx10}^* , C_{xxxx1}^* \right] \quad (5.138)$$

$$= \min \left[(siste\ best.\ i\ 4.\ periode) , (siste\ best.\ i\ 5.\ periode) \right] \quad (5.139)$$

$$= \min \left[(C_{xxx1}^* + h \cdot D_5) , (C_4^* + S) \right] \quad (5.140)$$

$$= \min \left[(10\ 300 + 10 \cdot 500) , (10\ 300 + 3100) \right] \quad (5.141)$$

$$= \min \left[15\ 300 , \underline{13\ 400} \right] \quad (5.142)$$

$$= \underline{13\ 400} \quad (5.143)$$

Vi har streket under for hvilken periode det er optimalt å ha siste bestilling, som i dette tilfellet er periode 5.

Vi kan dermed via Horisontteoremet kutte periode 4 fra videre beregninger.

¹⁰Siden er det 5 perioder så har C_{xxxxx} fem indekser i lign.(5.138).

Periode 6: (subproblem de 6 første periodene)

Vi starter dermed beregningene fra periode 5: ¹¹

$$\underline{C_6^*} = \min \left[C_{xxxx10}^* , C_{xxxx1} \right] \quad (5.144)$$

$$= \min \left[(siste\ best.\ i\ 5.\ periode) , (siste\ best.\ i\ 6.\ periode) \right] \quad (5.145)$$

$$= \min \left[(C_{xxx1}^* + h \cdot D_6) , (C_5^* + S) \right] \quad (5.146)$$

$$= \min \left[(13\ 400 + 10 \cdot 100) , (13\ 400 + 3100) \right] \quad (5.147)$$

$$= \min \left[\underline{14\ 400} , 16500 \right] \quad (5.148)$$

$$= \underline{14\ 400} \quad (5.149)$$

Vi har streket under for hvilken periode det er optimalt å ha siste bestilling, som i dette tilfellet er periode 5.

Vi kan dermed via Horisontteoremet kutte periode 4 fra videre beregninger (som vi allerede har gjort).

¹¹Siden er det 6 perioder så har C_{xxxxxx} seks indekser i lign.(5.144).

Periode 7: (subproblem de 7 første periodene)

Vi starter dermed beregningene fra periode 5: ¹²

$$\underline{C_7^*} = \left[C_{xxxx100}^*, C_{xxxx10}^*, C_{xxxxx1}^* \right] \quad (5.150)$$

$$= \min \left[(siste\ best.\ i\ 5.\ periode), (siste\ best.\ i\ 6.\ periode), (siste\ best.\ i\ 7.\ periode) \right]$$

$$= \min \left[(C_{xxx10}^* + 2 \cdot h \cdot D_7), (C_{xxxx1}^* + h \cdot D_7), (C_6^* + S) \right] \quad (5.151)$$

$$= \min \left[(14\,400 + 2 \cdot 10 \cdot 150), (16\,500 + 10 \cdot 150), (14\,400 + 3100) \right] \quad (5.152)$$

$$= \min \left[\underline{17\,400}, 18\,000, 17\,500 \right] \quad (5.153)$$

$$= \underline{17\,400} \quad (5.154)$$

Vi har streket under for hvilken periode det er optimalt å ha siste bestilling, som i dette tilfellet er fremdeles periode 5.

Vi kan dermed via horisontteoremet kutte periode 4 fra videre beregninger (som vi allerede har gjort).

¹²Siden er det 7 perioder så har C_{xxxxxx} syv indekser i lign.(5.150).

Periode 8: (det fulle problemet)

Vi starter dermed beregningene fra periode 5. Legg merke til at for uke 8 legger vi alltid til sluttlageret I_8 :¹³

$$\underline{C}_8^* = \min \left[C_{xxx\textcolor{red}{1}000}^* , C_{xxxx\textcolor{blue}{1}00}^* , C_{xxxxx\textcolor{green}{1}0}^* , C_{xxxxxx1}^* \right] \quad (5.155)$$

$$= \min \left[(s. b. i \textcolor{red}{5}. periode) , (s.b. i \textcolor{blue}{6}. periode) , (s. b. i \textcolor{green}{7}. periode) , (s. b. i 8. periode) \right]$$

$$= \min \left[\left(C_{xxx\textcolor{red}{1}00}^* + 3 \cdot h \cdot (D_8 + I_8) \right) , \left(C_{xxxx\textcolor{blue}{1}0^*} + 2 \cdot h \cdot (D_8 + I_8) \right) , \right. \\ \left. \left(C_{xxxxx\textcolor{green}{1}}^* + h \cdot (D_8 + I_8) \right) , (C_7^* + S) \right] \quad (5.156)$$

$$= \min \left[(\textcolor{red}{17\,400} + 3 \cdot 10 \cdot 500) , (\textcolor{blue}{18\,000} + 2 \cdot 10 \cdot 500) , \right. \\ \left. (\textcolor{green}{17\,500} + 10 \cdot 500) , (17\,400 + 3100) \right] \quad (5.157)$$

$$= \min \left[32\,400 , 28\,000 , 22\,500 , \underline{20\,500} \right] \quad (5.158)$$

$$= \underline{20\,500} \quad (5.159)$$

Vi har nå funnet optimale produksjonskostnader, som er

$$\underline{\underline{C^* = 20\,500}} \quad (5.160)$$

¹³Siden er det **8** perioder så har $C_{\textcolor{red}{xxxxxxx}}$ åtte indekser i lign.(5.155).

Vi fortsetter nå med å finne optimal [produksjonsplan](#),
dvs. vi skal bestemme Y_t , X_t og I_t -variablene for $t = 1, 2, 3, \dots, 8$.

Steg 4:

Vi følger steg 4 i algoritmen beskrevet på side 263 og starter bakerst og jobber oss fremover:

Delproblem 8:

Vi fant at:¹⁴

$$\begin{aligned}\underline{C}_8^* &= \min \left[(s. b. i 5. periode) , (s.b. i 6. periode) , (s. b. i 7. periode) , (s. b. i 8. periode) \right] \\ &= \min \left[32400, 28000, 22500 , \underline{20500} \right]\end{aligned}\tag{5.161}$$

Vi har derfor $Y_8 = 1$. Hittil har vi dermed funnet at optimal produksjonsplan er gitt ved

$$C_{xxxxxx1}\tag{5.162}$$

For å finne neste produksjonsperiode, må vi nå hoppe til delproblem 7:

¹⁴På eksamen og øvinger er det **ikke** nødvendig å skrive opp igjen disse utregningene - det er nok å skrive opp løsningen **direkte**, ved å lese av utregningene.

Delproblem 7:

Vi fant at:

$$\begin{aligned}\underline{C}_7^* &= \min \left[(siste\ best.\ i\ 5.\ periode) , (siste\ best.\ i\ 6.\ periode) , (siste\ best.\ i\ 7.\ periode) \right] \\ &= \min \left[\underline{17400} , 18000 , 17500 \right]\end{aligned}\tag{5.163}$$

Vi ser at neste produksjonsperiode¹⁵ er periode 5, dvs. $Y_5 = 1$, mens $Y_7 = 0$ og $Y_6 =$. Hittil har vi dermed funnet at optimal produksjonsplan er gitt ved

$$C_{xxxx1001}\tag{5.164}$$

For å finne neste produksjonsperiode, må vi nå hoppe til delproblem 4:

¹⁵Den optimale produksjonsperioden for det gitte delproblemet er understreket (huket av).

Delproblem 4:

Vi fant at:

$$\begin{aligned}\underline{C}_4^* &= \min \left[(\textit{siste best. i 2. periode}) , (\textit{siste best. i 3. periode}) , (\textit{siste best. i 4. periode}) \right] \\ &= \min \left[11200 , 12300 , \underline{10300} \right]\end{aligned}\tag{5.165}$$

Vi ser at neste produksjonsperiode er periode 4, dvs. $Y_4 = 1$. Hittil har vi dermed funnet at optimal produksjonsplan er gitt ved

$$C_{xxx11001}\tag{5.166}$$

For å finne neste produksjonsperiode, må vi nå hoppe til delproblem 3:

Delproblem 3:

Vi fant at:

$$\begin{aligned}\underline{C}_3^* &= \min \left[(\textit{siste best. i 2. periode}) , (\textit{siste best. i 3. periode}) \right] \\ &= \min \left[\underline{7200} , 9300 \right]\end{aligned}\tag{5.167}$$

Vi ser at neste produksjonsperiode er periode 2, dvs. $Y_2 = 1$. Hittil har vi dermed funnet at optimal produksjonsplan er gitt ved

$$C_{x1011001}\tag{5.168}$$

Delproblem 1:

For å finne neste produksjonsperiode, må vi nå hoppe til delproblem 1, men det problemet vet vi har $Y_1 = 1$. Optimale produksjonsperioder er dermed

$C_{11011001}$	(5.169)
----------------	---------

Algoritmen er nå **ferdig**. Vi kan nå oppsummere produksjonsplanen.

Optimale produksjonsperioder: $C_{11011001}$, dvs.:

$$\underline{\underline{Y_1 = 1}} \quad , \quad \underline{\underline{Y_2 = 1}} \quad , \quad \underline{\underline{Y_3 = 0}} \quad , \quad \underline{\underline{Y_4 = 1}} \quad (5.170)$$

$$\underline{\underline{Y_5 = 1}} \quad , \quad \underline{\underline{Y_6 = 0}} \quad , \quad \underline{\underline{Y_7 = 0}} \quad , \quad \underline{\underline{Y_8 = 1}} \quad (5.171)$$

Optimale seriestørrelser (dominante produksjonsplaner):

$$\underline{\underline{X_1}} \quad = \quad D_1 - I_0 \quad = \quad 300 - 50 \quad = \quad \underline{\underline{250}} \quad (5.172)$$

$$\underline{\underline{X_2}} \quad = \quad D_2 + D_3 \quad = \quad 400 + 100 \quad = \quad \underline{\underline{500}} \quad (5.173)$$

$$\underline{\underline{X_3}} \quad = \quad \underline{\underline{0}} \quad (5.174)$$

$$\underline{\underline{X_4}} \quad = \quad D_4 \quad = \quad \underline{\underline{200}} \quad (5.175)$$

$$\underline{\underline{X_5}} \quad = \quad D_5 + D_6 + D_7 \quad = \quad 500 + 100 + 150 \quad = \quad \underline{\underline{750}} \quad (5.176)$$

$$\underline{\underline{X_6}} \quad = \quad \underline{\underline{0}} \quad (5.177)$$

$$\underline{\underline{X_7}} \quad = \quad \underline{\underline{0}} \quad (5.178)$$

$$\underline{\underline{X_8}} \quad = \quad D_8 + I_8 \quad = \quad 400 + 100 \quad = \quad \underline{\underline{500}} \quad (5.179)$$

Optimale seriestørrelser (dominante produksjonsplaner):

$$\underline{\underline{I_1}} = \underline{\underline{0}} \quad (5.180)$$

$$\underline{\underline{I_2}} = D_3 = \underline{\underline{100}} \quad (5.181)$$

$$\underline{\underline{I_3}} = \underline{\underline{0}} \quad (5.182)$$

$$\underline{\underline{I_4}} = 0 \quad (5.183)$$

$$\underline{\underline{I_5}} = D_6 + D_7 = 100 + 150 = 250 \quad (5.184)$$

$$\underline{\underline{I_6}} = D_7 = \underline{\underline{150}} \quad (5.185)$$

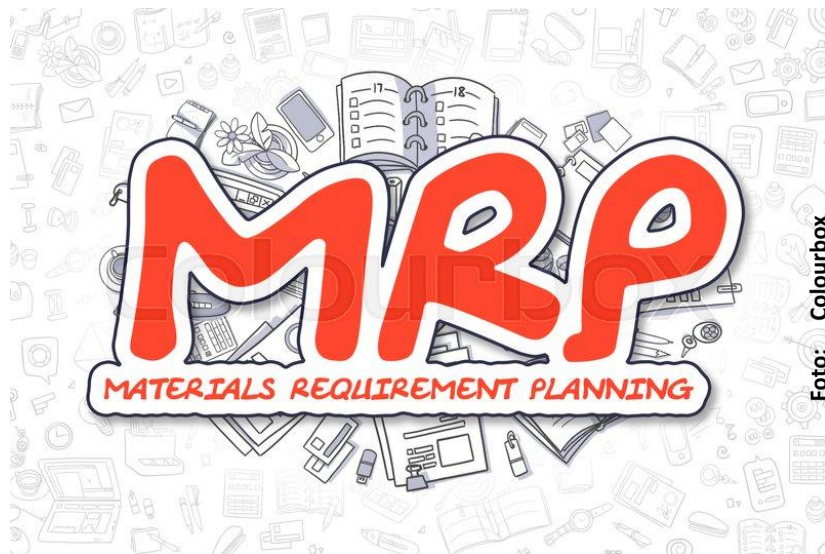
$$\underline{\underline{I_7}} = \underline{\underline{0}} \quad (5.186)$$

Gå tilbake på side 253 for å bekrefte at dette er samme løsning som vi fant ved hjelp av Lingo!



Kapittel 6

MRP - Material Requirements Planning



Figur 6.1: MRP.

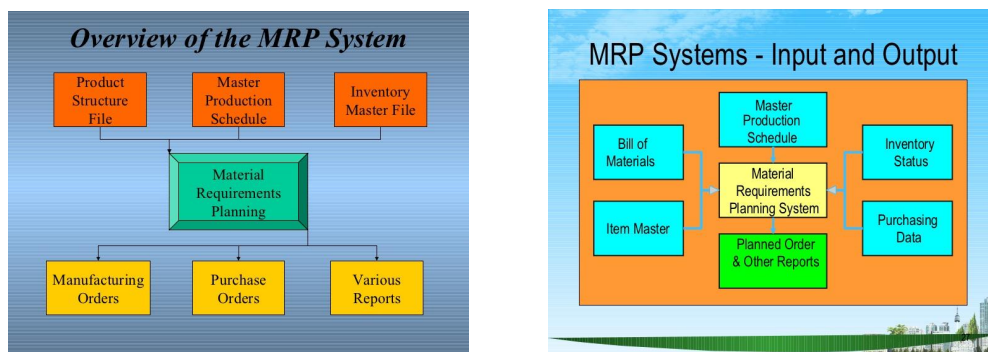
6.1 Introduksjon

I forrige kapittel gikk vi gjennom [master produksjonsplanlegging](#), hvor hovedresultatet var en [masterplan](#) for varer med [uavhengige etterspørsel](#) (kundestyrte).

I dette kapitlet skal vi se på fortsettelsen på masterplanene, hvor hvert produkt dekomponeres i komponenter via den såkalte [BOM-strukturen](#). BOM-strukturen viser hvordan en masterplanvare består av komponenter. Hver komponent kan igjen ha sin egen BOM-struktur osv. Denne hierarkiske oppsplittingen av komponenter kan fortsette i så mange nivåer som trengs inntil vi har råvarene som handles inn.

Etterspørselen for komponenter er [avhengig etterspørsel](#) siden etterspørselen kommer som en følge av masterplanen. Etterspørselen for komponenter er med andre ord avhengig av hvordan selve ferdigvarene er oppbygd.

MRP - eller *Material Requirements Planning* - omhandler planleggingen og styringen av hvordan vi skaffer inn de nødvendige komponentene slik at produksjonen av masterplanvarene kan gjennomføres som planlagt.



Figur 6.2: MRP illustrasjoner.

Desto mer komplisert BOM-strukturen er for en ferdigvare, jo vanskeligere blir MRP-planleggingen. Anskaffelsen av komponenter har alltid en [ledetid](#), som gjør at innkjøpsplanene/produksjonsplanene blir forskjøvet i tid i forhold til masterplanene. I tillegg er det ofte hensiktsmessig at ordrestørrelsene/seriestørrelsene for komponentene er større enn behovet fra masterplanen, siden vi da får flerdriftsfordeler (rabatter, lavere bestillingskostnader, mer kostnadseffektiv produksjon etc.). Disse kostnadene veies da opp mot lagerholdskostnaden for komponentene.

Å holde en MRP-plan for en komponent er kostbart og krever troverdige og gode data. Ikke alle komponenter eller råvarer har behov for å holde en MRP-plan. De minste komponentene som koster lite å holde på lager, har ikke behov for en komplisert MRP-plan. For å kunne bestemme hvilke artikler i bedriften det er hensiktsmessig å ha MRP-plan for, klassifiserer vi gjerne artiklene enten som A, B eller C artikler. Klassifiseringen kalles en [ABC-analyse](#), og hensikten er nettopp å dele inn artiklene gjennom [prosentvis andel av årlig omsetning](#). C-artiklene er typisk de små artiklene (som det er mange av) som ofte ikke har en MRP-plan knyttet til seg.

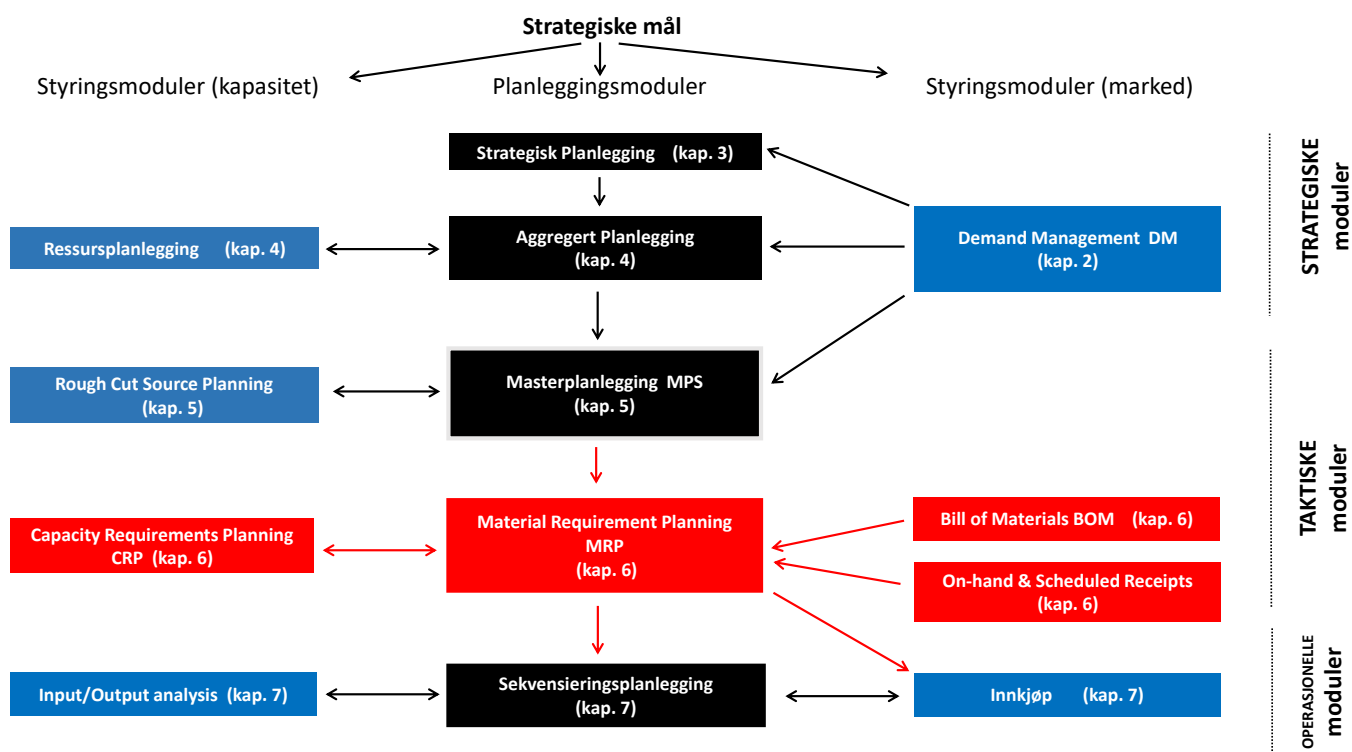
MRP algoritmen er en metode for å planlegge når og hvor mye vi skal anskaffe av hver artikkel og hvor mye vi skal ha på lager. Algoritmen er [suboptimal](#) med tanke på f.eks. kostnader, men er likevel hovedkjernen i ethvert ERP-system på grunn av algoritmens beregningstid samt at den søker å minimere lagerholdet (dvs. at vi timer inn slik at komponentene ligger kortest mulig tid på lageret). Algoritmens største problem er at den ikke tar høyde for kapasiteter i bedriften.

Verdensbildet: ([globalt](#))

MRP-modulen er en taktisk modul som blant annet henter BOM-data fra BOM-modulen. Siden metoden ikke tar høyde for kapasiteter, har vi en egen modul som beregner og planlegger kapasitetsforbruket: *Capacity Requirements Planning* (CPR). CPR-modulen kan medføre replanlegging av MRP-planene inntil kapasitetsforbruket er innenfor grensene.

I tillegg får MRP-modulen styringsdata fra *On-hand receipts & scheduled receipts*-modulen. Dette er varer under arbeid, dvs. en tilstandsrapport som sender tilstandsdata til MRP-modulen. Store avvik fra MRP-planen, kan medføre replanlegginger.

Fra MRP-planene genereres en såkalt "job pool", dvs. en liste av jobber som skal gjennomføres. Noen av disse jobbene medfører direkte innkjøp (råvarer/3.parts komponenter), mens andre komponenter sendes for allokering til produksjonen.



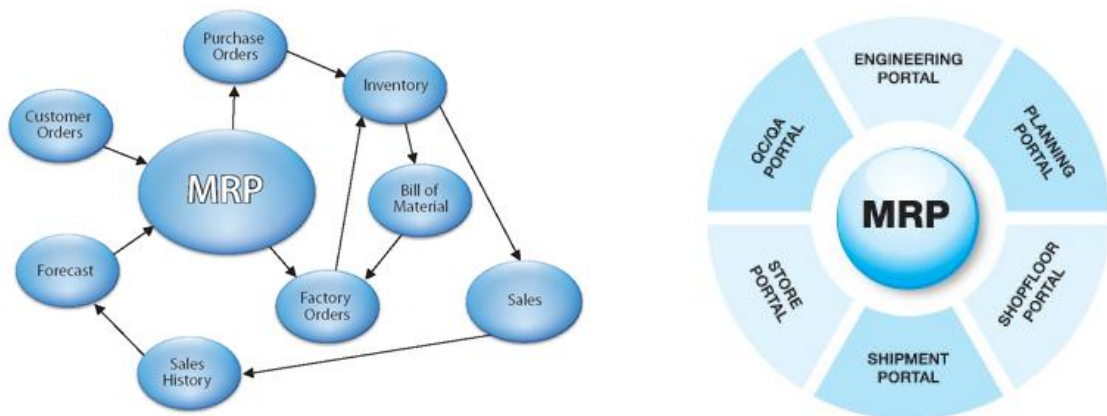
Figur 6.3: Moduler for et MPC-system - verdensbildet.

6.2 Hovedaktiviteter

MRP-modulen svarer i prinsippet på akkurat de samme beslutningene som Master produksjonsplanlegging, det eneste som endrer seg er at vi nå ser på komponenter og råvarer:

- **hvilke** komponenter holder vi MRP-plan for?
- **hvor mye** av hver komponent skal vi anskaffe?
- **hvor mye** av hver komponent skal vi ha på lager?
- **når** skal komponentene være på lager?

På styringssiden handler det først og fremst om å **oppdatere/replanlegge** MRP-planene. På grunn av usikkerhet både ved etterspørsel og leveranser, vil det alltid forekomme endringer som ofte krever manuelle beslutninger fra planleggerne.



Figur 6.4: MRP illustrasjoner.

Typiske aktiviteter ved MRP-planlegging: (aktiviteter)

Typiske planleggingsaktiviteter:

- innhente masterplaner
- bestemme hvor mye skal vi produsere/bestille (ordrestørrelser) og hvor mye skal vi ha på lager for hver periode
- bestemme hvordan etterspørsel skal oppfylles, dvs. om vi skal tillate stockout eller ikke
- bestemme servicegrad
- bestemme hvorvidt vi skal holde sikkerhetslager, og hvor stort det eventuelt skal være

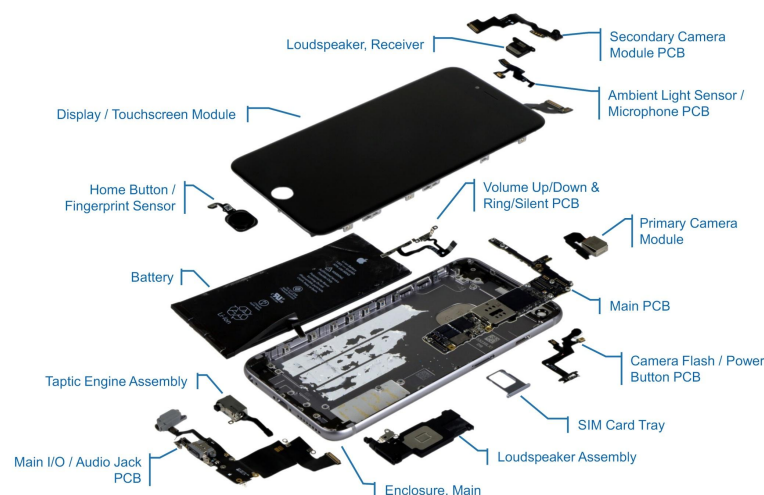
6.3 BOM-strukturer

BOM-modulen er en egen modul hvor bedriften legger inn hvordan en ferdigvare er oppbygd av komponenter. Disse komponentene kan igjen være oppbygd av nye komponenter. Slik kan vi fortsette inntil vi kun har råverer igjen (innkjøpsvarer).

BOM - *Bill Of Materials* - fremstår i mange fasonger, her er to av dem:

- grafisk fremstilt
- liste med innrykk

For hver komponent, påpekes hvor mange enheter av komponenten som inngår i [farskomponenten](#).



Figur 6.5: BOM for iPhone 6S.

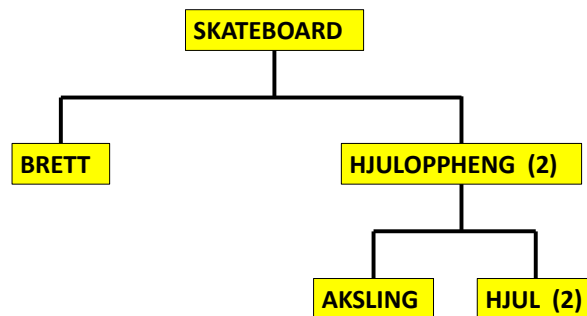
Eksempel: (BOM-figurer)

Et firma som produserer skateboard trenger følgende komponenter: brett (1 stk.), hjuloppheng (2 stk.), hjul (2 stk.), aksel (1 stk.)

Lag en grafisk fremstilling av BOM-strukturen.

Lag også en liste med innrykk.

Løsning:



Figur 6.6: Grafisk fremstilling av BOM-struktur.

Komponent	Antall som behøves
Skateboard	1
Brett	1
Hjuloppheng	2
Hjul	2
Aksling	1

Figur 6.7: Listefremstilling av BOM-struktur.



6.4 MRP-modellen

Den generelle MRP-modellen er en ganske komplisert modell på grunn av faseforskyvninger siden

- ledetider inkluderes
- den hierarkiske oppbygningen av et produkt (BOM)

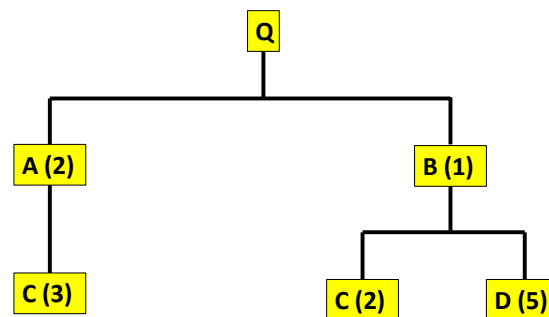
Vi beskriver derfor heller en spesifikk MRP-modell gjennom et eksempel.

Eksempel: ([MRP-modell](#))

Vi skal i denne oppgaven beregne MRP-planer for et produkt Q med komponenter A , B , C og D .

Anta at produktet Q har BOM-struktur som vist i figur 6.8.

Planleggingshorisonten er 10 uker.



Figur 6.8: BOM-strukturen for produkt Q .



Foto: Colourbox

Figur 6.9: BOM.

Bruttobehovet for produktet Q kommer fra masterplanen for dette produktet. Legg merke til at poster med verdien 0 ikke skrives inn (for oversiktens skyld).

Uke t	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov D_t		20	20	20	10	10	20		40		40

Tabell 6.1: Bruttobehov fra masterplanen.

Legg merke til:

En masterplan gir behov for når ferdigvarene er på **lager**, ikke når de må bestilles. Dette gir en **faseforskyvning** mellom masterplan og tilhørende MRP-planer som en må ta høyde for ledetiden for produkt Q (produksjonstiden).

Fra BOM-strukturen i figur 6.9 ser vi at produktet Q består av totalt 4 komponenter: A , B , C og D . Nødvendige data er oppgitt i tabell 6.2.

Komponent	Ledetid	Bestillingskost	Lagerkost	På lager	Mottak
Q	2	3000	400	50	-
A	3	2000	300	75	-
B	1	1000	400	35	-
C	2	500	10	100	300 i uke 1
D	2	700	20	20	-

Tabell 6.2: Data for produkt Q og dets komponenter.

Mottak er i prinsippet varer i arbeid som er på vei til lager. Det er nettopp her modulen *On-hands & scheduled receipts* benyttes.

- a) Beskriv data i modellen.
- b) Beskriv variabler i modellen
- c) Beskriv målfunksjonen i modellen.
- d) Beskriv føringene i modellen.

Beskrive en matematisk modell som skal minimere totale lager- og bestillingskostnader.

Løsning:

1) Data:

Siden denne modellen er nokså stor, må vi benytte indeksnotasjon. Vi har **to** indekser for dette problemet:

- **komponentene**, som indekseres ved $i = Q, A, B, C, D$.
- **periodene**, som indekseres ved $t = 1, 2, \dots, 10$.

$$I_{i0} = \text{initielt lager inn til uke 1 for komponent } i \quad (6.1)$$

$$I_{i10} = \text{sluttlager inn til uke 10 for komponent } i \quad (6.2)$$

$$D_t = \text{etterspørsel for komponent } Q \text{ i uke } t, \text{ hvor } t = 1, 2, \dots, 10 \quad (6.3)$$

$$h_i = \text{overlagringskostnad per enhet for komponent } i \quad (6.4)$$

$$S_i = \text{bestillingskostnad for komponent } i \quad (6.5)$$

$$L_i = \text{ledetiden for komponent } i \quad (6.6)$$

$$M_{it} = \text{antall enheter som er planlagt mottak for komponent } i \text{ for periode } t \quad (6.7)$$

Merk:

Fra tabell 6.2 ser vi at det kun er $M_{C1} = 300$ som er større enn null. De øvrige M_{it} er null.

2) Variable:

Vi har tre typer variabler i problemet:

- **ordrestørrelse** - hvor mange enheter vi skal bestille/produsere pr periode
- **lagernivå** - hvor mange enheter vi skal ha på lager fra en periode til neste
- **bestille/ikke bestille** - ja/nei variabel som avgjør om vi skal ha bestilling/produksjon eller ikke

a) **Ordrestørrelser/seriestørrelser:**

$$X_{it} = \text{antall enheter som kommer inn p lager av komponent } i \text{ for periode } t \quad (6.8)$$

hvor $t = 1, 2, \dots, 10$.

b) **Lagernivå:**

$$I_{it} = \text{antall enheter overlagret av komponent } i \text{ fra periode } t \text{ til periode } t + 1 \quad (6.9)$$

hvor $t = 1, 2, \dots, 10$.

c) **Bestille/ikke bestille** (samme for produksjon):

$$Y_t = \begin{cases} 1 & , \text{ dersom vi bestiller/produserer i periode } t \\ 0 & , \text{ hvis ikke} \end{cases} \quad (6.10)$$

hvor $t = 1, 2, \dots, 8$.

3) Målfunksjon:

Totale kostnader fås ved å summere bestillingskostnadene/setupkostnadene og lagerkostnadene for alle periodene i horisonten:

$$C = \sum_{i=X}^D \sum_{t=1}^{10} S_i Y_t + \sum_{i=X}^D \sum_{t=1}^{10} h I_t \quad (6.11)$$

4) Føringer:

Vi har to føringer i modellen:

- a) **lagerbalansen** - som sørger for at all etterspørsel oppfylles
- b) **logisk sammenheng** mellom ordrestørrelse (X_t -variablene) og hvorvidt vi bestiller eller ikke (Y_t -variablene)

a) **Lagerbalansen:**

Det er her ved **lagerbalansen** at de store forskjellene ift. masterplanleggingen skjer. Vi må håndtere flere endringer:

- Vi må følge BOM-strukturen, dvs. **etterspørselen** for en gitt komponent er lik **summen** produksjonsstørrelsene for alle **farskomponentene**.
- Vi må ta høyde for at hver bestilling har en **ledetid**.

Komponent Q :

$$I_{Qt-1} + X_{Qt-2} - D_{1t} = I_{Qt} \quad (6.12)$$

som er lagerbalanse for periode t , hvor $t = 1, 2, \dots, 10$.

Komponent A:

$$I_{At-1} + X_{At} - 2X_{Qt+2} = I_{At} \quad (6.13)$$

som er lagerbalanse for periode t , hvor $t = 1, 2, \dots, 8$.

Komponent B:

$$I_{Bt-1} + X_{Bt} - X_{Qt+2} = I_{Bt} \quad (6.14)$$

som er lagerbalanse for periode t , hvor $t = 1, 2, \dots, 8$.

Komponent C:

$$I_{C0} + M_{C1} + X_{C1} - 3X_{A3} - 2X_{B3} = I_{C1} \quad (6.15)$$

som er lagerbalanse for periode 1, og videre

$$I_{Ct-1} + X_{Ct} - 3X_{At+3} - 2X_{Bt+1} = I_{Ct} \quad (6.16)$$

som er lagerbalanse for periode t , hvor $t = 1, 2, \dots, 5$.

Komponent D :

$$I_{Dt-1} + X_{Dt} - 5X_{Bt+1} = I_{Dt} \quad (6.17)$$

som er lagerbalanse for periode t , hvor $t = 1, 2, \dots, 7$.

b) **Logisk sammenheng:**

Vi kan uttrykke føringen *logisk* ved:

$$\boxed{\text{Hvis } Y_{it} = 0 \text{ så må } X_{it} = 0} \quad (6.18)$$

for hver periode t , hvor $t = 1, 2, \dots, 10$.

Alternativt kan vi skrive denne føringen på matematisk ikke-linær form:

$$\boxed{X_{it}(1 - Y_{it}) = 0} \quad (6.19)$$

for hver periode t , hvor $t = 1, 2, \dots, 10$, eller på linær form:

$$\boxed{X_{it} \leq MY_{it}} \quad (6.20)$$

for hver periode t , hvor $t = 1, 2, \dots, 10$, hvor $M = \sum_{t=1}^T D_t$ er summen av all etterspørsel i horisonten.

■

Vi har nå formulert MRP-modellen. Som vanlig ønsker vi nå å løse modellen.

MRP-modellen er dessverre for komplisert til at Lingo finner løsning (for mer reelle problemstillinger).

I de to neste kapitlene går vi gjennom MRP-algoritmen, som er en tilnærming som søker å minimere lageret. Metoden er den mest tradisjonelle metoden brukt i alle ERP-systemer, og ligger fundamentalt i ethvert produksjonsplanleggingssystem.

Vi starter med å introdusere MRP-tabellen, før vi går gjennom selve algoritmen.

6.5 MRP-tabellen

I tabellen under har vi oppgitt en ferdig utregnet MRP-tabell for komponent Q fra eksempelet over. Tabellen representerer [planen](#) for de neste 10 ukene for komponent Q .

Uke		Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov	D_{Qt}		20	20	20	10	10	20		40		40
Planlagt mottak	M_{Qt}											
Lager	I_{Qt}	50	30	10								
Nettobehov	X_{Qt}				10	10	10	20		40		40
Planlagt ordre	X_{Qt-L_Q}		10	10	10	20		40		40		

Tabell 6.3: MRP-tabell for produkt Q .

En MRP-tabell består av følgende felter (for hver periode):

- [Bruttobehov](#) - det [totale](#) behovet for komponenten (avhengig etterspørsel). Siden en bil har fire felger, kan vi anta at det er planlagt å produsere 25 biler i 3. 4. 6. og 8. uke henholdsvis.
- [Planlagt mottak](#) - bekreftede ordrer som skal leveres i løpet av horisonten (samme som for masterplan)
- [Sikkerhetslager](#) - dersom komponenten opererer med sikkerhetslager, oppgis dette her. (NB: denne raden finnes ikke for reelle MRP-tabeller, men inkluderes her for læringens skyld).
- [Lager](#) - planlagte lagernivåer i løpet av horisonten
- [Nettobehov](#) - det [faktiske](#) behovet for komponenten, etter at lager og eventuelle planlagte mottak har blitt lagt til.
- [Planlagt ordre](#) - produksjons -eller innkjøpsplaner. Tallene viser når varene må bestilles, dvs [ledetiden](#) for anskaffelsen av bestillingen er tatt med i betraktningen. Hvilken ordrestørrelse som benyttes antas gitt når vi beregner MRP-planene. Seriestørrelsene er dog en helt fundamental beslutning, og man kan benytte de samme metodene som for masterplanlegging (EOQ, Wagner-Whitin etc).

Datasystemet selv kan endre planlagte ordrer, siden posten ikke representerer ressurser som er tatt i bruk.

Når en ordre faktisk blir plassert, flyttes den fra planlagt ordre til planlagt mottak.

Datasystemet kan da ikke endre ordren, kun manuell overstyring kan finne sted.

MRP-planene er også [rullerende planer](#), som er tilfellet for masterplanene også. Når en periode har passert, oppdateres MRP-tabellene automatisk, hvor planen forskyves en uke frem i tid.

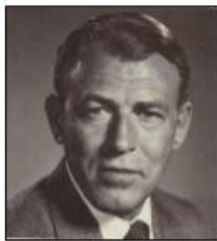
6.6 MRP-algoritmen (planleggingsmetodikken)

I 1964 utviklet [Joe Orlicky](#) en [algoritme](#) for å beregne MRP-planer for komponentene ved produksjon/innkjøp. Denne algoritmen ble selve startmodulen i dagens ERP-systemer. MRP-modulen er således den eldste modulen i verdensbildet.

I tiden før MRP-algoritmen ble hovedsaklig EOQ og bestillingspunktmodellen brukt.

MRP-algoritmen skal i prinsippet finne en [tilnærmet](#) løsning på MRP-problemet fra avsnitt 6.4.

1975: The First Significant Explanation of MRP



Joe Orlicky



*"As this book goes into print, there are some **700 manufacturing companies or plants** that have implemented, or are committed to implementing, MRP systems. Material requirements planning has become a new way of life in production and inventory management, displacing older methods in general and statistical inventory control in particular. I, for one, have no doubt whatever that it will be the way of life in the future."*

Figur 6.10: Joseph Orlicky (1922 - 1986).

Data til MRP-algoritmen:

- bruttobehovet for mastervarene (fås fra MPS-tallene i masterplanen)
- BOM-strukturene for hver komponent (inkludert mastervaren)
- ledetider for hver komponent (innkjøp/produksjon)
- eventuelle sikkerhetslager for hver komponent
- initielt lager inn til første periode i planen
- planlagte ordrer (varer i arbeid)
- seriestørrelser for hver komponent (MPS-størrelser)

I tillegg kan det være flere andre data som er aktuelle, som for eksempel svinn på produksjon/lager etc.

Legg merke til at [seriestørrelsene](#) allerede er bestemt. Dette er X -variablene i MRP-modellen. Dvds. at vi fastsetter seriestørrelser før vi i det hele tatt starter algoritmen. Disse seriestørrelsene fås gjennom tradisjonell EOQ, Wagner-Whitin e.l. for hver enkelt komponent.

Algoritme: (MRP-algoritmen , planleggingsmetodikken)

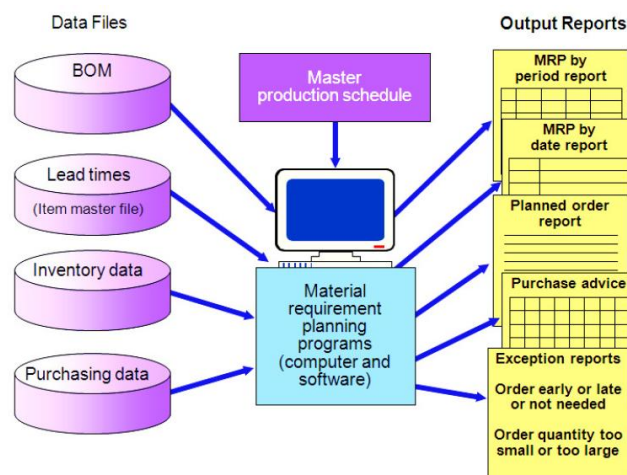
MRP-algoritmen er en algoritme som løser planleggingsproblemet for avhengige komponenter, med mål om å *minimere* kostnader.

Algoritmen følger følgende:

1. Start med en masterplan vare. Bruttobehovet for denne varen er gitt fra masterplanen, og da spesifikt MPS-størrelsen. Beregn MRP-tabellen for denne varen.
2. Finn BOM-strukturen for komponentene. Velg en subkomponent (hvis flere igjen)
3. Beregn bruttobehovet for komponenten. [Bruttobehovet fås ved å legge til alle planlagte ordrer for alle farskomponentene](#) (legg merke til at planlagte ordrer for alle farskomponentene må allerede være beregnet)
4. Beregn lagernivå for hver uke ved hjelp av [lagerbalanselikningen](#)¹. Hvis lagerbalansen ikke går opp, genereres et nettobehov, som igjen trigger en planlagt ordre. Den planlagte ordren må forskyves i forhold til ledetiden.
5. Gå til punkt 2 hvis komponenten har delkomponenter (dvs. komponenten har sin egen BOM-struktur).

Når det ikke er flere delkomponenter igjen, er algoritmen ferdig.

■



Figur 6.11: MRP-algoritmen.

¹Lagerbalansen er [eneste](#) korrekte måte å regne ut lagernivået på hvis man er usikker.

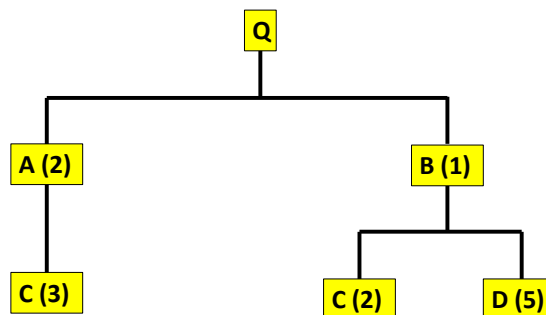
Kommentarer til algoritmen:

- Legg merke til at algoritmen ikke beregner [kostnadene](#). Istedet forsøker algoritmen å minimere [lagernivået](#), ved å vente lengst mulig før en ordre legges inn. Algoritmen er således [suboptimal](#).
- Algoritmen tar [ikke kapasitetsføringer i betraktning](#). Istedet sendes MRP-planen til en egen modul kalt CRP-modulen (Capacity Requirements Planning). Her beregnes kapasitetsbehovet for den gitte MRP-planen. Hvis planen ikke er gjennomførbar, må rutinen kjøres på nytt med nye data (manuelle endringer).

Eksempel: ([MRP-beregninger - fortsettelse fra eksempel på side 289](#))

Vi skriver for oversiktens skyld opp igjen hele beskrivelsen av eksempelet fra side 289.

Vi skal i denne oppgaven beregne MRP-planer for et produkt Q hvor planleggingshorisonten er 10 uker.
Anta at produktet Q har BOM-struktur som vist i figur 6.12.
Dette er samme figuren som figur 6.8.



Figur 6.12: BOM-strukturen for produkt Q .

Bruttobehovet for produktet Q kommer fra masterplanen for dette produktet. Legg merke til at poster med verdien 0 ikke skrives inn (for oversiktens skyld). Tabellen nedenfor, dvs. tabell 6.1, er samme tabell som vist i tabell 6.4.

Uke t	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov D_t		20	20	20	10	10	20		40		40

Tabell 6.4: Bruttobehov fra masterplanen.

Fra BOM-strukturen i figur 6.12 ser vi at produktet Q består av totalt 4 komponenter:

A , B , C og D .

Nødvendige data er oppgitt i tabell 6.5. **Lot-For-Lot** betyr at vi bestiller akkurat likt nettobehovet (metoden er listet opp under lagerstyring i kapittel 5 om master planlegging). Vi har fjernet lagerkostnadene og bestillingskostnadene fra tabellen, siden MRP-algoritmen ikke bruker disse tallene.

Komponent	Ledetid	Seriestørrelse	På lager	Mottak
Q	2	Lot-For-Lot	50	-
A	3	100	75	-
B	1	50	35	-
C	2	300	100	300 i uke 1
D	2	300	20	-

Tabell 6.5: Data for produkt Q .

- a) Beregn og fyll inn verdiene for MRP-tabellen for produkt Q .
- b) Beregn og fyll inn verdiene for MRP-tabellen for komponent A .
- c) Beregn og fyll inn verdiene for MRP-tabellen for komponent B .
- d) Beregn og fyll inn verdiene for MRP-tabellen for komponent C .
- e) Beregn og fyll inn verdiene for MRP-tabellen for komponent D .

Løsning:

a) Produkt Q :

Produkt Q er top-level komponenten.

Vi har fått oppgitt bruttobehovene. Vi har ikke noe planlagt mottak for denne komponenten. Initielt lager er 50 enheter og vi bruker Lot-For-Lot ordrestørrelse:

Uke t	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov D_t		20	20	20	10	10	20		40		40
Planlagt mottak											
Lager I_t	50	30	10								
Nettobehov				10	10	10	20		40		40
Planlagt ordre		10	10	10	20		40		40		

Tabell 6.6: Data for produkt Q .

Vi ser for eksempel at nettobehovet i uke 3 er satt som en planlagt ordre **to uker før** (som er ledetiden), dvs. i uke 1.

b) Komponent A:

BOM-strukturen i figur 6.12 viser at vi må ha 2 komponenter av A for hvert produkt av Q .

Siden MRP-algoritmen venter så lenge som mulig med å legge inn bestillinger, vil 2*(planlagte ordrer) for produkt Q bli bruttobehovet for komponent A . Vi har 75 enheter initielt på lager og ledetid 3 uker. Ordrestørrelsen er på 100 enheter.

Uke t	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov D_t		20	20	20	40		80		80		
Planlagt mottak											
Lager I_t	75	55	35	15	75	75	95	95	15	15	15
Nettobehov					25		5				
Planlagt ordre		100		100							

Tabell 6.7: Data for komponent A .

c) Komponent B :

BOM-strukturen viser at vi må ha en komponent av B for hvert produkt av Q . Siden MRP-algoritmen venter så lenge som mulig med å sette bestillinger, vil planlagte ordrer for Q bli bruttobehovet for B . Vi har 35 enheter på lager og ledetid på en uke. Ordrestørrelsen er på 50 enheter.

Uke t	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov D_t		10	10	10	20		40		40		
Planlagt mottak											
Lager I_t	35	25	15	5	35	35	45	45	5	5	5
Nettobehov					15		5				
Planlagt ordre				50		50					

Tabell 6.8: Data for komponent B .

d) Komponent C :

BOM-strukturen viser at vi må ha 3 komponenter av C for hver komponent av A . I tillegg trenger vi to komponenter av C for hver komponent av B . Vi må legge sammen disse behovene for å få det totale bruttobehovet for komponent C . Vi har et planlagt mottak på 300 enheter av komponent C i uke 1. Vi har 100 enheter på lager og ledetid på to uker. Ordrestørrelsen er på 300 enheter.

Fra komponent A får vi to behov:

- Uke 1 ordre på 100 A -komponenter gir 300 C -komponenter
- Uke 3 ordre på 100 A -komponenter gir 300 C -komponenter

Fra komponent B får vi to behov:

- Uke 3 ordre på 50 B -komponenter gir 100 C -komponenter
- Uke 5 ordre på 50 B -komponenter gir 100 C -komponenter

Siden vi har to ordrer i uke 3 blir det totale bruttobehovet for komponent C i uke 3 lik summen: $300 + 100 = 400$.

Uke t	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov D_t		300		400		100					
Planlagt mottak		300									
Lager I_t	100	100	100			200	200	200	200	200	200
Nettobehov				300		100					
Planlagt ordre		300		300							

Tabell 6.9: Data for komponent C .

e) Komponent D :

BOM-strukturen viser at vi må ha 5 komponenter av D for hver komponent av B . Siden MRP-algoritmen venter så lenge som mulig med å sette bestillinger, vil bruttobehovet for komponent D bli $5 \cdot (\text{planlagte ordrer})$ for komponent B . Vi har 20 enheter på lager og ledetid på to uker. Ordrestørrelsen er på 300 enheter.

Fra komponent B får vi to behov:

- Uke 3 ordre på 50 B -komponenter gir 250 C -komponenter
- Uke 5 ordre på 50 B -komponenter gir 250 C -komponenter

Uke	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov				250		250					
Planlagt mottak											
Lager	20	20	20	70	70	120	120	120	120	120	120
Nettobehov				230		180					
Planlagt ordre		300		300							

Tabell 6.10: Data for komponent D .

Siden vi nå har beregnet MRP-tabellene for alle komponentene, er algoritmen ferdig.

■

Eksempel: (sikkerhetslager i MRP-tabeller)

Vi antar nå at vi bestemmer oss for å holde et **sikkerhetslager** på 10 enheter for komponent B fra forrige eksempel.

Sikkerhetslager er et lager som skal ta høyde for usikkerhetsfaktorer som for eksempel variasjoner i bruttobehovet eller variasjoner i ledetidene.

Helt fundamentalt for et sikkerhetslager er at det i prinsippet **alltid skal være der**, dvs det ideellt sett skal aldri røres.

Oppdater MRP-tabellen for komponent B når sikkerhetslager er inkludert.

Løsning:

Beste måte å behandle sikkerhetslager på er å introdusere en ny rad i MRP-tabellen kalt "Sikkerhetslager". Tanken er at vi **initielt** deler opp det totale lageret i to deler: sikkerhetslager og planlagt lager. Når den initielle oppdelingen er gjort, kan vi fylle ut **alle** feltene for sikkerhetslageret, slik at dette er uberørt. Vi beregner da det resterende lageret som før ved hjelp av lagerbalansen.

Vi splitter det initielle lageret bestående av 35 enheter i 10 enheter for sikkerhetslager og 25 enheter for planlagt lager.

Uke t	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov D_t		10	10	10	20		40		40		
Planlagt mottak											
Sikkerhetslager	10	10	10	10	10	10	10	10	10	10	10
Lager I_t	25	15	5	45	25	25	35	35	45	45	45
Nettobehov				5			15		5		
Planlagt ordre			50			50		50			

Tabell 6.11: Sikkerhetslager for komponent B .



Eksempel: ([svinn i MRP-tabeller](#))

Vi antar nå produksjonen for komponent D har et svinn (skrap) på 10 %.

Vesentlig for skrap er at bestillingene følger ordrestørrelsen men faktiske enheter som kommer inn på lageret er redusert med 10 %.

Oppdater MRP-tabellen for komponent D når man tar **hensyn til svinn**.²

Løsning:

Siden ordrestørrelsen er på 300 enheter, får vi et ukentlig svinn på 30 enheter. Vi må altså oppdatere lageret med 270 enheter og ikke 300 enheter som vi har bestilt.

Uke t	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov D_t				250		250					
Planlagt mottak											
Lager I_t	20	20	20	40	40	90	40	90	90	90	90
Nettobehov				230		210					
Planlagt ordre		300		300							

Tabell 6.12: Data for omponent D .

Lagerbalansen for uke 3 blir: $20 + 300 \cdot 0.9 - 250 = 40$.

■

²Vi bruker den opprinnelige tabellen for komponent B, dvs. ikke den som tar høyde for sikkerhetslageret.

Eksempel: (rullerende horisont - MRP)

MRP-tabellene er rullerende, dvs. at når en uke har passert, så går denne uken ut, hele planene forskyves en uke og inn kommer en ny uke i uke 8.

Det initielle lageret oppdateres i forhold til det faktiske salget.

Anta vi ser på komponent X, hvor en uke har passert og det faktiske salget ble på 15 enheter (det forventede salget var på 20 enheter). Anta at det nye bruttobehovet i uke 8 er null.

Oppdater MRP-tabell for produkt Q etter at en uke har passert.

Løsning:

Siden det faktiske salget i uke 1 ble 15, sitter vi igjen med et initielt lager på 35 enheter. I tillegg har vi fått et planlagt mottak fra den planlagte ordren vi hadde i uke 1.

Uke t	Init	1	2	3	4	5	6	7	8	9	10
Bruttobehov D_t		20	20	10	10	20		40		40	
Planlagt mottak		10									
Lager I_t	35	25	5								
Nettobehov				5	10	20		40		40	
Planlagt ordre		5	10	20		40		40			

Tabell 6.13: Oppdatert MRP-tabell for produkt Q .



Vi har nå gått gjennom MRP-algoritmen og hvordan den håndterer typiske situasjoner som sikkerhetslager og svinn i produksjonen.

I neste kapittel samler vi alle ordrene og sender dem til det [operasjonelle](#) nivået hvor komponentene og produktet faktisk blir produsert/bestilt. Men først skal vi gå gjennom en viktig [klassifisering](#) av lagerartikler, som f.eks. hjelper oss med å fortelle [hvilke](#) artikler vi skal holde MRP-planer for.

6.7 ABC-analyse

MRP-beregninger er kostbare å holde. For at MRP-tabellene skal ha nytte, må dataene som benyttes være av høy kvalitet. Det er derfor ikke hensiktsmessig å holde MRP-tabeller for absolutt alle lagerartiklene som holdes i bedriften.

Vi har behov for å [kategorisere](#) lagerartiklene. [ABC-analyse](#) er nettopp en slik kategorisering, hvor artiklene deles inn i tre klasser: klasse *A*, klasse *B* og klasse *C*.

Siden det som regel er [lagerkostnaden](#) som er den viktigste faktoren for lagerholdet, baseres en ABC-analyse på [årlig andel av total lageromsetning](#).

Klassifiseringen er som følger:

- A-artiklene utgjør totalt 70 % av total lageromsetning
- B-artiklene utgjør totalt 25 % av total lageromsetning
- C-artiklene utgjør totalt 5 % av total lageromsetning

hvor alle artiklene er [sortert etter økende andel total omsetning](#).

Som regel er det slik i en bedrift at relativt få artikler utgjør 70 % av total lageromsetning, mens svært mange artikler utgjør kun 5 % av omsetningen. Typisk er den velkjente "[Paretos prinsipp](#)" (80-20-regelen) oppfylt: 20 % av artiklene utgjør 80 % av omsetningen.

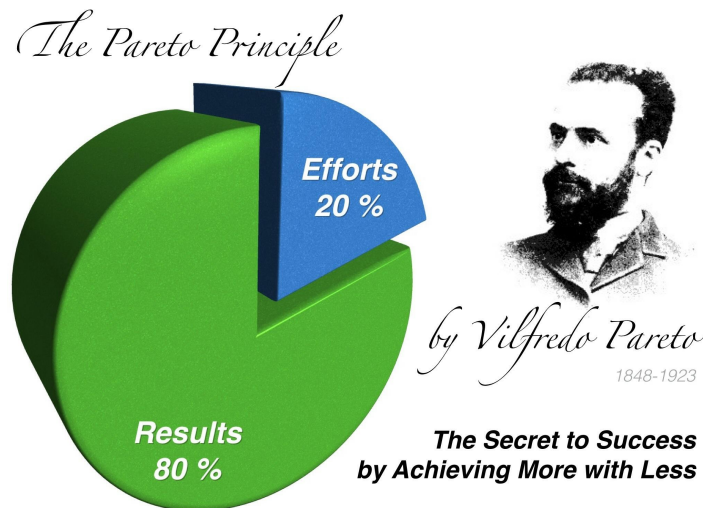
Prinsipp: (Paretos prinsipp , også kalt 80-20 regelen) ³

Paretos prinsipp er observasjonen at innsats og resultat ikke er likt fordelt, og ofte 80-20 fordelt:

20 % av innsatsen gir 80 % av resultatet

Paretos prinsipp kan f.eks. bety:

- 20 % av artiklene gir 80 % av omsetningen (logistikk)
- 20 % av kundene gir 80 % av resultatet (økonomi)
- 20 % av feilene gir 80 % av datakræsjene (software)



Figur 6.13: Paretos prinsipp.

³Paretos prinsipp er ikke en naturlov. Det er et prinsipp basert på observasjoner.

Eksempel: (ABC-klassifisering)

Vi skal i denne oppgaven gjennomføre en ABC-klassifisering på ti av artiklene til Brunvoll AS.

Vi har oppgitt følgende data for hver artikkel:

Varenr	Årlig forbruk (enheter)	Enhetspris
1	6	90
2	78	250
3	7	300
4	3	220
5	21	50
6	8	90
7	10	30
8	40	120
9	6	30
10	3	50

Tabell 6.14: Data.

Vi starter ABC-analysen med å beregne årlig omsetning for hver artikkel:

Varenr	Årlig forbruk (enheter)	Enhetspris	Årlig omsetning
1	6	90	540
2	78	230	17940
3	7	300	2100
4	3	220	660
5	21	50	1050
6	8	90	720
7	10	30	300
8	40	80	3200
9	6	30	180
10	3	50	150

Tabell 6.15: Data og tilhørende omsetning.

Deretter må vi **sortere** tabellen etter **minkende årlig omsetning**, dvs. de artiklene med størst omsetning først osv.

Varenr	Årlig forbruk (enheter)	Enhetspris	Årlig omsetning
2	78	230	17940
8	40	80	3200
3	7	300	2100
5	21	50	1050
6	8	90	720
4	3	220	660
1	6	90	540
7	10	30	300
9	6	30	180
10	3	50	150

Tabell 6.16: Sortering etter minkende årlig omsetning.

Deretter regner vi ut kumulert omsetning i % og samtidig klassifiserer vi produktene på følgende måte:

- A-produkter: 0-70 %
- B-produkter: 70-95 %
- C-produkter: 95-100 %

I grensetilfellene hvor et produkt spenner over skillet fra en klasse til en annen, er det rom for skjønn. Vi velger som regel den klassen som artikkelen ligger *mest* innenfor.

Varenr	Årlig forbruk	Enhetspris	Årlig omsetning	kum.oms. i %	Klasse
2	78	230	17940	66.8	A
8	40	80	3200	78.8	A
3	7	300	2100	86.6	B
5	21	50	1050	90.5	B
6	8	90	720	93.2	B
4	3	220	660	95.6	B
1	6	90	540	97.7	C
7	10	30	300	98.8	C
9	6	30	180	99.4	C
10	3	50	150	100	C

Tabell 6.17: Klasse *A*, *B* og *C*.

Hvis vi husker Paretos prinsipp, som sier at 20 % av produktene utgjør 80 % av omsetningen, så ser vi at dette er tilfelle her hvor 78.8 % av total omsetning dekkes av 20 % av artiklene (dvs. 2 av 10 artikler).

■

Kapittel 7

Operasjonell planlegging



Foto: Colourbox

Figur 7.1: Operasjonell planlegging.

7.1 Introduksjon

Vi har nå kommet til det **operasjonelle** nivået i planleggingshierarkiet.

Fra MRP-modulen får vi en **job-pool** som skal gjennomføres i løpet av den gitte planleggingshorisonten - som nå gjerne spenner seg fra 1 til 2 uker. I stedet for uker, ser vi på dager og timer og i svært repetetive situasjoner også minutter.

Hva er det som **mangler** i MRP-planen?

Vi mangler **hvor** og **tidspunkt** for når jobbene (MPS-ordrene) fra MRP-modulen faktisk skal gjennomføres.



Figur 7.2: Operasjonell planlegging.

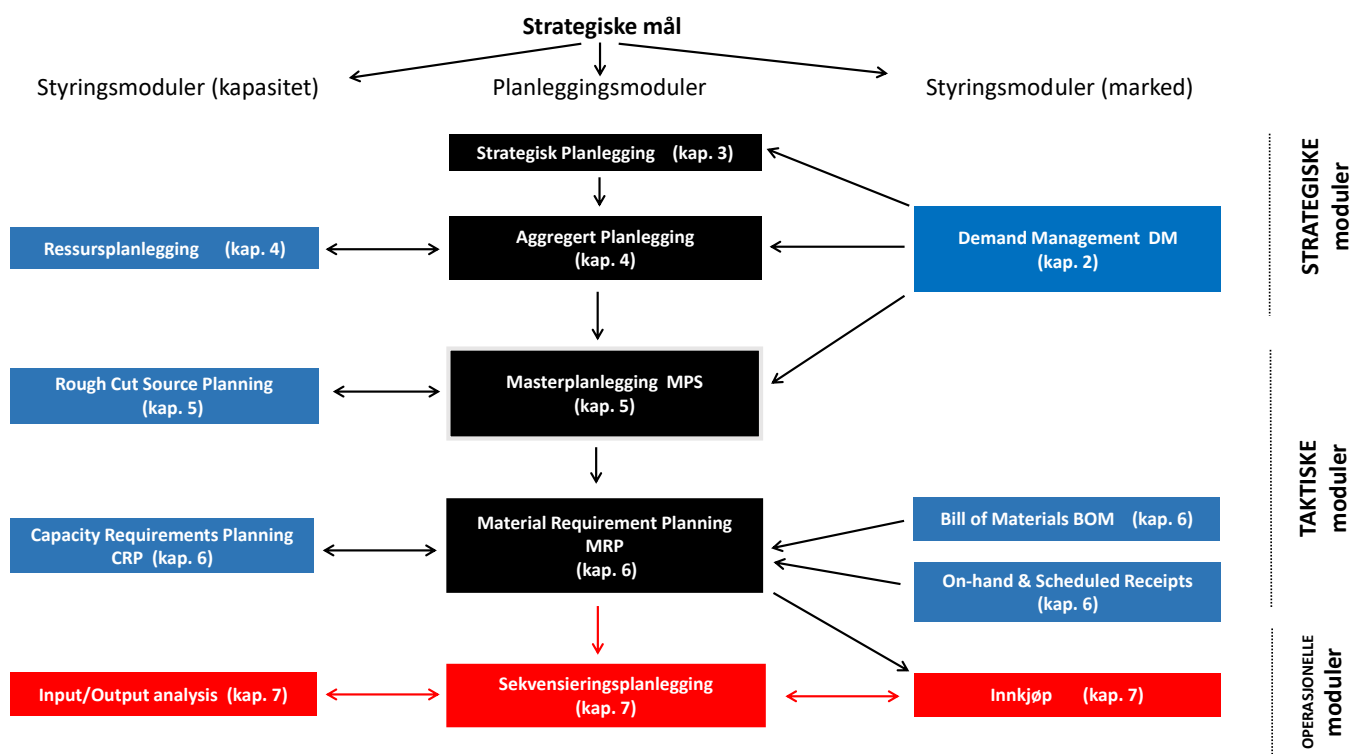
To sentrale begreper i kapitlet:

- **Jobb** - en faktisk komponent som skal **bearbeides** i produksjonen. En jobb har:
 - Et visst antall delkomponenter (BOM)
 - Et visst antall **operasjoner** som må gjennomføres, gjerne i en gitt rekkefølge
 - Ofte en **due-date** - dvs. siste tidspunkt for ferdigstilling.
- **Operasjon** - et enkeltsteg i en jobb, som gjerne er knyttet til en maskin og/eller operatør. En operasjon har:
 - En **bearbeidingstid** (prosesseringsstid)
 - Et planlagt starttidspunkt (dato og klokkeslett)
 - Et planlagt sluttstidspunkt (dato og klokkeslett)
 - Ofte en **due-date** - dvs. siste tidspunkt for ferdigstilling.

Verdensbildet: (globalt)

Vi ser på hovedsaklig tre av modulene (innkjøpsmodulen og leverandørsystemer går ikke gjennom i dette kurset):

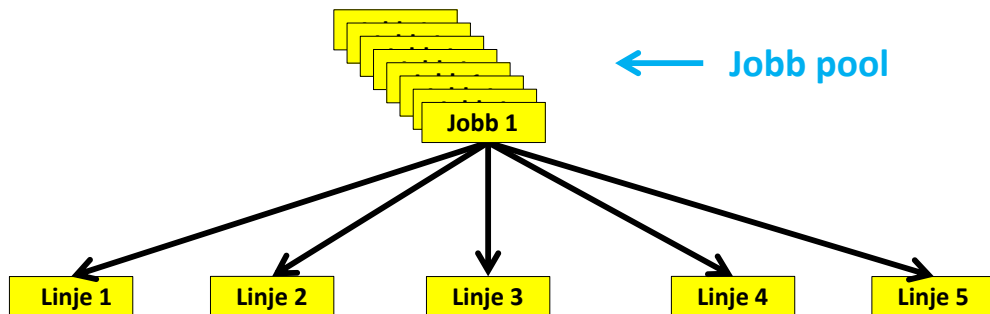
- **Jobb allokering** - (job release) - her mottar vi jobbene fra MRP-modulene og **allokerer** dem til linje/arbeidsstasjon/fasilitet.
- **Jobb sekvensiering** - (job scheduling) - her fastsetter **timeplanen**, dvs. start- og sluttidspunkter for alle operasjonene som skal gjennomføres for hver linje/arbeidsstasjon/fasilitet.
- **Input/output analysis** - styringsmodul hvor vi kontinuerlig måler avvik i forhold til planen. modulen kan trigge replanlegginger (vi går heller ikke nærmere inn på denne modulen).



Figur 7.3: Moduler for et MPC-system - verdensbildet.

7.2 Jobb allokering

Jobb allokering handler om å bestemme hvilken linje/arbeidsstasjon/fabrikk hver jobb skal tilordnes til.



Figur 7.4: Jobb allokering - hvilke linjer skal ta hvilke jobber.

Analogi:

Anta du vil ha en burger på McDonalds, hvor det er tre skranke som tar imot ordrer. Anta det er køforan alle skrankene.

Hvilken skranke velger du?

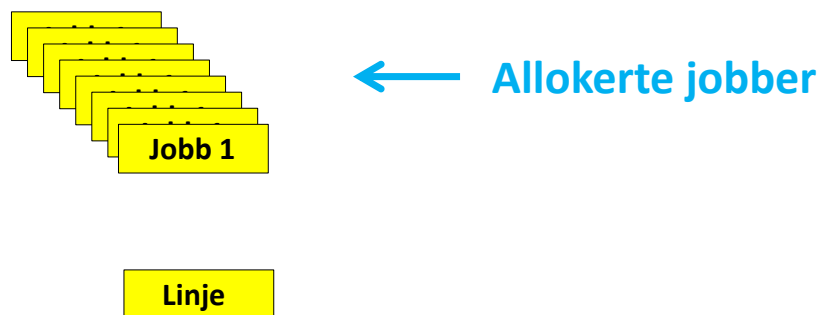
Som regel velges den skranken hvor det er antatt minst ventetid.

■

7.3 Sekvensiering

Sekvensiering handler om å bestemme **riktig rekkefølge** på jobber og operasjoner for en gitt linje/arbeidsstasjon¹.

Vi antar nå at vi har allokert et visst antall jobber til en linje/arbeidsstasjon. Siden jobbene ofte har forskjellige prosesseringstider, vil **rekkefølgen** vi prosesserer jobbene og operasjonene på ha **stor innvirkning** på hvordan produksjonen gjennomføres.



Figur 7.5: Sekvensiering - rekkefølgen et sett med jobber prosesseres.

Men hva mener vi med ”stor innvirkning”?

For å kunne snakke om gode og dårlige sekvensieringsplaner, må vi ha en målfunksjon som vi kan evaluere planene med.

¹”Jobb” og ”operasjon” brukes ofte om hverandre når det gjelder allokering og sekvensiering, siden det er svært situasjonsbetinget hvordan en bedrift definerer **sine** jobber og operasjoner. Spesielt hvis en jobb kun består av en operasjon, er begrepene helt sammenfallende.

7.3.1 Målfunksjoner

Målfunksjoner: ([makespan](#) og [gjennomsnittlig gjennomløpstid](#))

Produksjonsstyring og spesielt sekvensering har svært mange distinkte målfunksjoner. Å velge rett målfunksjon er ikke alltid enkelt, og for sekvensering vil valget av målfunksjon være helt essensielt.

Her er noen populære målfunksjon:

- [Makespan](#) - tidspunkt for når siste operasjon er ferdig
- [Gjennomsnittlig gjennomløpstid](#) - hvor lenge hver jobb er under prosessering i gjennomsnitt
- [antall forsinkede jobber](#) - hvor mange jobber som er forsinket i forhold til due date
- [total forsinkelse](#) - summen av all forsinkelse

Vi skal se nærmere på de to av første:

Makespan og gjennomsnittlig gjennomløpstid, siden disse målfunksjonene representerer to motsette målsetninger.

Makespan: ([tidspunkt for siste ferdigstillelse](#))

[Makespan](#) står for [tidspunkt for siste ferdigstillelse](#). Makespan brukes som målfunksjon når vi ønsker å utnytte [kapasiteten](#) best mulig.

Kapasitet er et vanskelig begrep, og det er enda vanskeligere å maksimere utnyttelsen av kapasitet. Problemet ligger i mangfoldigheten i hvordan kapasitet beskrives. Det mest vanlige målet på kapasitet er hvor mye vi får produsert i løpet av en viss tidsperiode. Jo mer vi får produsert under en gitt tidsperiode, desto bedre utnytter vi tilgjengelig kapasitet.

Anta vi har et sett med jobber som skal kjøres gjennom en produksjonslinje. Hvis vi minimerer makespan, vil vi, siden makespan er tidspunkt for siste ferdigstillelse, samtidig presse ferdigstillelsen for alle de [andre](#) ordrene innen dette tidspunktet. Vi får dermed gjort så mye som mulig på kortest mulig tid. Dette innebærer for eksempel at arbeidsstasjonene går minst mulig ledig. På denne måten kan vi si at minimering av makespan vil medføre best mulig kapasitetsutnyttelse.

Makespan brukes dermed ofte som målfunksjon i situasjoner hvor kapasitetsutnyttelse er i størst fokus. Verftsindustrien er et typisk eksempel på dette, siden de opererer med høye fastkostnader som maskinnedskriving og lønnskostnader (kostnader som ikke kan påvirkes under operasjonell planlegging).

Gjennomsnittlig gjennomløpstid: ([gjennomsnittlig tid i prosessering](#))

[Gjennomsnittlig gjennomløpstid](#) måler hvor lenge hver ordre er under prosessering og tar deretter gjennomsnittet av disse.

Anta for eksempel at vi har prosessert tre jobber, som tok henholdsvis 20, 15 og 25 timer. Gjennomsnittlig gjennomløpstid er det gitt ved

$$\frac{20 + 15 + 25}{3} = \frac{60}{3} = 20 \text{ timer} \quad (7.1)$$

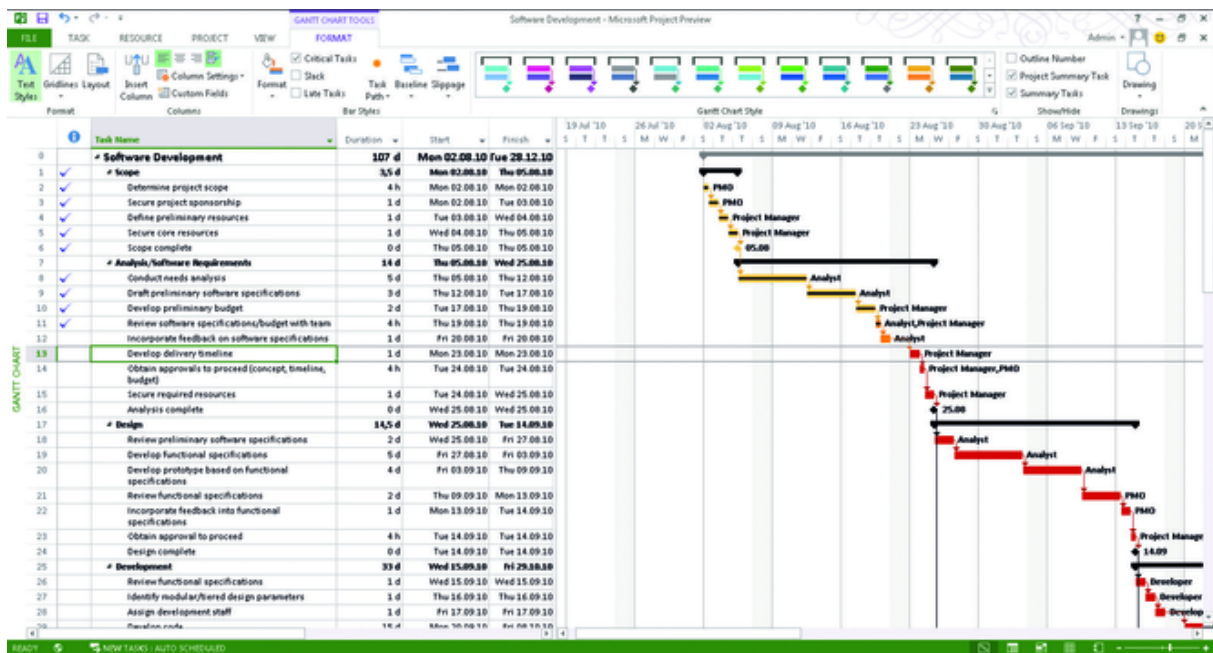
Å minimere gjennomsnittlig gjennomløpstid vil ikke nødvendigvis bety effektiv kapasitetsutnyttelse, men kan derimot være forbundet med lavere [kostnader](#).

Hvis det er forbundet en eller flere merkostader med hver jobb mens jobben er under prosessering, er gjennomsnittlig gjennomløpstid en ofte brukt målfunksjon. Ta for eksempel pasienter som behandles på sykehus - her er tida på sykehuset forbundet med en kostnad, og dermed vil gjennomsnittlig "liggetid" på sykehuset være interessant å forbedre.

7.3.2 Gantt-diagrammer

Hvilke verktøy brukes ved jobb sekvensiering?

Figur 7.6, viser et [Gantt-diagram](#), som er et essensielt verktøy når man jobber med sekvensiering.



Figur 7.6: Gantt-diagram fra MS Project

Gantt-diagrammet viser på en svært visuell måte når hver enkelt operasjon skal gjennomføres.

Et Gantt-diagram kan ha forskjellige detaljniva:

- Forskjellig tidsskala fra dager til timer til minutter
- Kan også vises pr. linje eller pr. maskin

7.4 Sekvensieringscase: En stasjon med 6 ordrer

Eksempel: (en stasjon med 6 ordrer)

Vi skal nå se på et eksempel hvor vi har en stasjon og 6 ordrer som skal prosesseres gjennom en arbeidsstasjon. Anta at jobbene ble allokert i alfabetisk rekkefølge.

Vi har oppgitt følgende data:

Jobb	Tidsfrist (due time)	Bearbeidingstid
A	3	4
B	9	2
C	4	1
D	15	5
E	11	3.5
F	19	4

Tabell 7.1: Data.

Oppgaven er å vurdere 5 forskjellige sekvensieringsregler mtp. **totalt antall forsinkede jobber og gjennomsnittlig forsinkelse**:

1. **FIFO** - First In First Out - ikke nødvendigvis bra siden tidlig ankommede jobber kan vente til fordel for jobber som haster mer.
2. **Due date** - vi ordner jobbene etter due date, dvs vi starter med den jobben som skal være ferdig først. Regelen er ofte brukt ved MRP bedrifter (produksjonsmiljø).
3. **Korteste jobb** - gir ofte få men store forsinkelser. Metoden er problematisk hvis nye jobber ankommer, siden de lange jobbene blir stadig mer utsatt.
4. **Korteste slakk** - velg de med minst slakk først. Slakk er differansen mellom due time og gjestående tid. Gir ofte små men mange forsinkelser.
5. **Kritisk ratio** - regner ut forholdet mellom due date og bearbeidingstid, og velger de med laveste verdi (kritisk ratio) først. Metoden fremstår ofte som fornuftig, siden den tar høyde for flere faktorer samtidig.

- a) Finn totalt antall forsinkelser og gjennomsnittelig forsinkelse ved bruk av **FIFO**.
- b) Som oppgave **a** - men for “**due date**”.
- c) Som oppgave **a** - men for “**korteste jobb**”.
- d) Som oppgave **a** - men for “**korteste slakk**”.
- e) Som oppgave **a** - men for “**kritisk ratio**”.

a) FIFO:

Vi beregner starttidspunkt, sluttidspunkt og forsinkelse:

Jobb	Due	Bearb.tid	Start	Slutt	Forsinkelse
A	3	4	0	4	1
B	9	2	4	6	
C	4	1	6	7	3
D	15	5	7	12	
E	11	3.5	12	15.5	4.5
F	19	4	15.5	19.5	0.5

Tabell 7.2: FIFO.

$$\underline{\underline{\text{totalt antall forsinkelser} = 4 \text{ jobber}}} \quad (7.2)$$

$$\underline{\underline{\text{gj.snittlig forsinkelse}}} = \frac{1 + 3 + 4.5 + 0.5}{6} = \frac{9}{6} = \underline{\underline{1.5 \text{ timer}}} \quad (7.3)$$

b) Due date:

Vi sorterer ordrene etter due date og beregner starttidspunkt, sluttidspunkt og forsinkelse:

Jobb	Due	Bearb.tid	Start	Slutt	Forsinkelse
A	3	4	0	4	1
C	4	1	4	5	1
B	9	2	5	7	
E	11	3.5	7	10.5	
D	15	5	10.5	15.5	0.5
F	19	4	15.5	19.5	0.5

Tabell 7.3: Due date.

$$\underline{\underline{\text{totalt antall forsinkelser}}} = 4 \text{ jobber} \quad (7.4)$$

$$\underline{\underline{\text{gj.snittlig forsinkelse}}} = \frac{1 + 1 + 0.5 + 0.5}{6} = \frac{3}{6} = \underline{\underline{0.5 \text{ timer}}} \quad (7.5)$$

c) Korteste jobb:

Vi sorterer ordrene etter bearbeidingstid og beregner starttidspunkt, sluttidspunkt og forsinkelse:

Jobb	Due	Bearb.tid	Start	Slutt	Forsinkelse
C	4	1	0	1	
B	9	2	1	3	
E	11	3.5	3	6.5	
A	3	4	6.5	10.5	7.5
F	19	4	10.5	14.5	
D	15	5	14.5	19.5	4.5

Tabell 7.4: Kortest bearbeidingstid.

$$\underline{\underline{\text{totalt antall forsinkelser} = 2 \text{ jobber}}} \quad (7.6)$$

$$\underline{\underline{\text{gj.snittlig forsinkelse}}} = \frac{7.5 + 4.5}{6} = \frac{12}{6} = \underline{\underline{2 \text{ timer}}} \quad (7.7)$$

d) Slakk:

Vi beregner og sorterer ordrene etter slakk (differansen mellom due date og bearbeidingstid) og beregner starttidspunkt, sluttidspunkt og forsinkelse:

Jobb	Due	Bearb.tid	Slakk	Start	Slutt	Forsinkelse
A	3	4	-1	0	4	1
C	4	1	3	4	5	1
E	11	3.5	6.5	5	8.5	
B	9	2	7	8.5	10.5	1.5
D	15	5	10	10.5	15.5	0.5
F	19	4	15	15.5	19.5	0.5

Tabell 7.5: Minst slakk først

$$\underline{\underline{\text{totalt antall forsinkelser} = 5 \text{ jobber}}} \quad (7.8)$$

$$\underline{\underline{\text{gj.snittlig forsinkelse}}} = \frac{1 + 1 + 1.5 + 0.5 + 0.5}{6} = \frac{4.5}{6} = \underline{\underline{0.75 \text{ timer}}} \quad (7.9)$$

Legg merke til at vi får mange, men korte forsinkelser.

e) Kritisk ratio:

Vi beregner og sorterer ordrene etter kritisk ratio (due date/bearbeidingstid) og beregner start-tidspunkt, sluttidspunkt og forsinkelse:

Jobb	Due	Bearb.tid	Slakk	Kritisk ratio	Start	Slutt	Forsinkelse
A	3	4	-1	0.75	0	4	1
D	15	5	10	3	4	9	
E	11	3.5	7.5	3.14	9	12.5	1.5
C	4	1	3	4	12.5	13.5	9.5
B	9	2	5	4.5	13.5	15.5	6.5
F	19	4	15	4.75	15.5	29.5	0.5

Tabell 7.6: Kritisk ratio

$$\underline{\underline{\text{totalt antall forsinkelser} = 5 \text{ jobber}}} \quad (7.10)$$

$$\underline{\underline{\text{gj.snittlig forsinkelse}}} = \frac{1 + 1.5 + 9.5 + 6.5 + 0.5}{6} = \frac{19}{6} = \underline{\underline{3.166 \text{ timer}}} \quad (7.11)$$

■

7.5 Sekvensieringscase: Sykehus

Vi skal nå se på et eksempel hvor vi har **to stasjoner** og **to ordrer** som skal prosesseres gjennom begge stasjonene i serie.

Eksempel: (**to ordrer** - to maskiner)

Asmund og Bård skal gjennom de samme to avdelingene på Molde sykehus.

Først skal de til generell poliklinisk undersøkelse og deretter skal de til operasjonsstua. Begge to møter opp klokka **08:00** til sykehuset.

Bård, som har en del idrettsskader, har estimert en tid på **5 timer** på undersøkelse og **2 timer** til operasjon.

maskin 1
maskin 2

Asmund, som er plaget med rygg, må ha **3 timer** til undersøkelse og **4 timer** til operasjon.

maskin 1 maskin 2

Vurder om Asmund skal prosesseres før Bård eller omvendt.

Vurdere dette med tanke på både **makespan** og **gjennomsnittlig gjennomløpstid**.

Sett opp Gantt-diagrammer for begge alternativene.

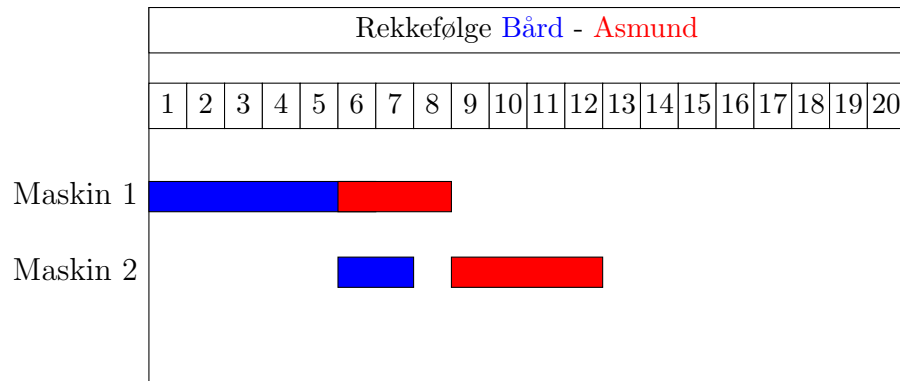


Foto: Øystein Bjerkeland

Figur 7.7: Molde sykehus.

Løsning:

1) Gantt-diagram for situasjonen Bård før Asmund:



Figur 7.8: Gantt-diagram.

Fra Gantt-diagrammet ser vi at:

$$\underline{\underline{\text{makespan}}} = \text{sluttid Asmund} = \underline{\underline{12 \text{ timer}}} \quad (7.12)$$

og

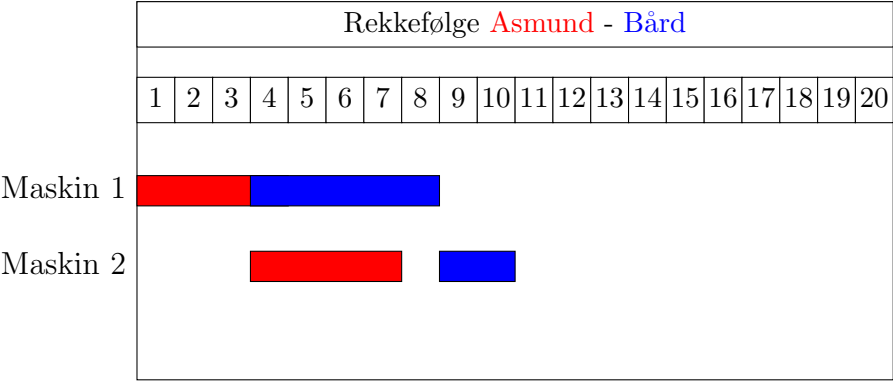
$$\begin{aligned} \underline{\underline{\text{total gjennomløpstid}}} &= \text{sluttid Asmund} + \text{sluttid Bård} \\ &= \left(12 + 7 \right) \text{ timer} = \underline{\underline{19 \text{ timer}}} \end{aligned} \quad (7.13)$$

samt (deler på 2 siden det er to stk. ordrer)

$$\underline{\underline{\text{gjennomsnittlig gjennomløpstid}}} = \frac{\text{total gjennomløpstid}}{2} \quad (7.14)$$

$$= \frac{19}{2} \text{ timer} = \underline{\underline{9.5 \text{ timer}}} \quad (7.15)$$

2) Gantt-diagram for situasjonen Asmund før Bård:



Figur 7.9: Gantt-diagram.

Fra Gantt-diagrammet ser vi at:

$$\underline{\text{makespan}} = \text{sluttid Bård} = \underline{\underline{10 \text{ timer}}} \tag{7.16}$$

og

$$\begin{aligned} \underline{\text{total gjennomløpstid}} &= \text{sluttid Asmund} + \text{sluttid Bård} \\ &= \left(7 + 10 \right) \text{ timer} = \underline{\underline{17 \text{ timer}}} \end{aligned} \tag{7.17}$$

samt (deler på 2 siden det er to stk. ordrer)

$$\underline{\underline{\text{gjennomsnittlig gjennomløpstid}}} = \frac{\text{total gjennomløpstid}}{2} \tag{7.18}$$

$$= \frac{17}{2} \text{ timer} = \underline{\underline{8.5 \text{ timer}}} \tag{7.19}$$

Konklusjon:

Vi konkluderer med at

Asmund bør prosesseres før Bård

mtp. både makespan, total gjennomløpstid og gjennomsnittlig gjennomløpstid.



7.6 Sekvensieringscase: To maskiner

Eksempel: (to maskiner)

En bedrift har 2 ordrer A og B som skal tilvirkes på 2 maskiner: maskin 1 og 2.

Ordre A bruker 3 timer på maskin 1 og 5 timer på maskin 2, mens ordre B bruker 4 timer og 2 timer på maskin 1 og 2.

Begge ordrene må bearbejdes først på maskin 1 og deretter på maskin 2.

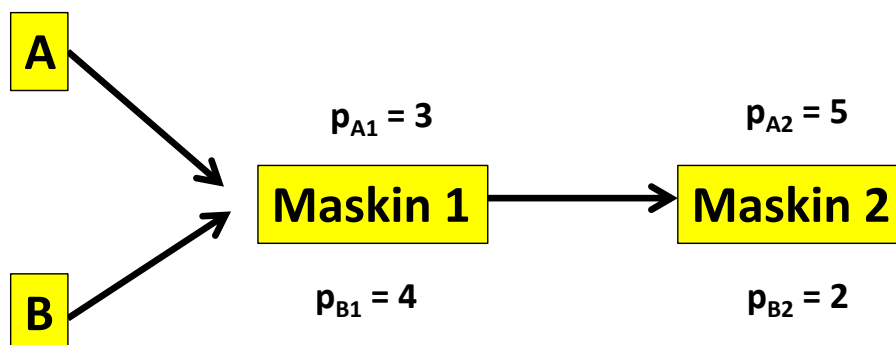


Figur 7.10: To maskiner.

- a) Finn minimal gjennomløpstid og minimalt makespan for sekvensieringsproblemet ved å sette opp Gantt-diagrammer for de forskjellige alternativene, les av verdiene og velg den beste sekvensen.
- b) Formuler dette planleggingsproblemet som et matematisk programmeringsproblem og løs det ved hjelp av Lingo med de samme målfunksjonene som i oppgave a. Bekreft at de optimale verdiene er samsvarende med de du fant fra oppgave a.
- c) Anta at bedriften har fått inn en ny ordre C før tilvirkningen av A og B har startet. Bedriften ønsker å replanlegge sekvensieringen slik at ordre C er tatt med i beregningen. Ordre C bruker 1 time på maskin A og 5 timer på maskin B .
- i) Løs replanleggingsproblemet som i oppgave a, dvs. ved hjelp av Gantt-diagram.
- ii) Utvid det matematiske programmeringsproblemet slik at ordre C er inkludert.

Løsning:

- a) Vi tegner opp en graf som viser flyten for de to ordrene A og B gjennom maskin 1 og deretter maskin 2.



Figur 7.11: To maskiner og to jobber.

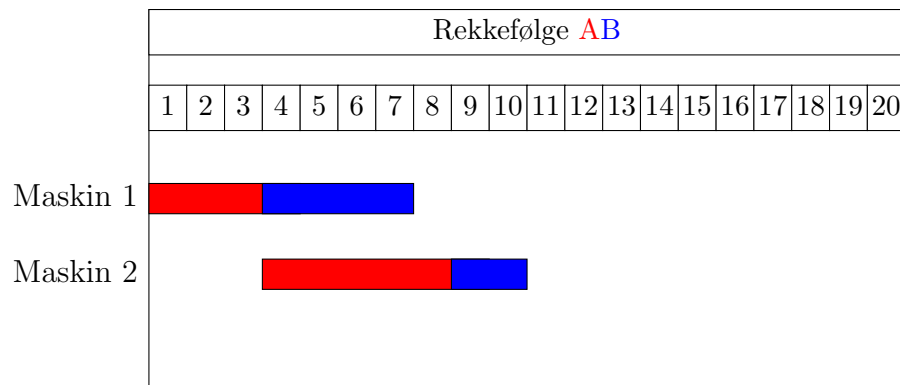
Det første vi spør oss, er hvor mange *kombinasjoner* finnes det?

Siden vi har to ordrer og to maskiner får vi $2^2 = 4$ muligheter, men i prinsippet er kun *to* av dem aktuelle, siden hvis A prosesseres først på maskin1, så må også A prosesseres først på maskin 2. Å splitte ordrekkefølgen på maskinene er åpenbart ineffektiv, så vi ser vekk ifra disse.

Vi har dermed *to* muligheter: AB eller BA.

Gantt-diagram for situasjonen AB, dvs. A først:

Merk: Rød er ordre A mens blå er ordre B:



Figur 7.12: Rekkefølge AB.

Fra Gantt-diagrammet:

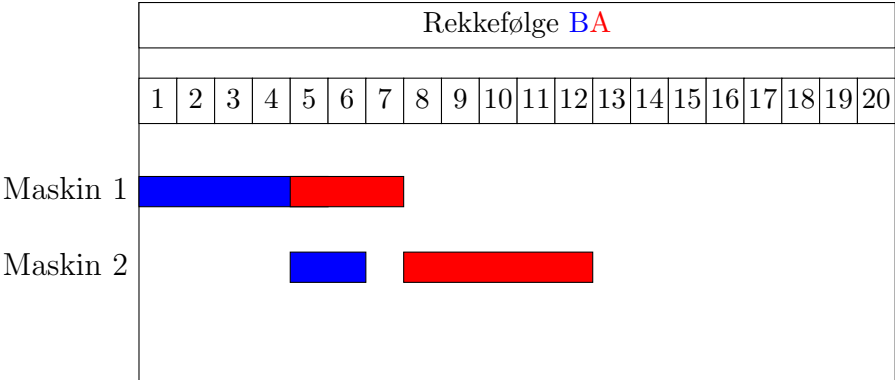
$$\underline{\underline{\text{makespan}}} = \text{sluttid } B = \underline{\underline{10 \text{ timer}}} \quad (7.20)$$

$$\underline{\underline{\text{total gjennomløpstid}}} = \text{sluttid } A + \text{sluttid } B = 8 + 10 = \underline{\underline{18 \text{ timer}}} \quad (7.21)$$

$$\underline{\underline{\text{gjennomsnittlig gjennomløpstid}}} = \frac{\text{total gjennomløpstid}}{2} = \frac{18}{2} = \underline{\underline{9 \text{ timer}}} \quad (7.22)$$

Gantt-diagram for situasjonen BA, dvs. B først:

Merk: Rød er ordre A mens blå er ordre B:



Figur 7.13: Rekkefølge BA.

Fra Gantt-diagrammet:

$$\underline{\underline{\text{makespan}}} = \text{sluttid } A = \underline{\underline{12 \text{ timer}}} \tag{7.23}$$

$$\underline{\underline{\text{total gjennomløpstid}}} = \text{sluttid } A + \text{sluttid } B = 6 + 12 = \underline{\underline{18 \text{ timer}}} \tag{7.24}$$

$$\underline{\underline{\text{gjennomsnittlig gjennomløpstid}}} = \frac{\text{total gjennomløpstid}}{2} = \frac{18}{2} = \underline{\underline{9 \text{ timer}}} \tag{7.25}$$

Konklusjon:

Vi ser at hvis vi ønsker å minimere makespan, så er det optimalt med rekkefølgen AB, mens hvis vi bruker total gjennomløpstid så er både AB OG BA optimale.

b) Vi skal nå lage en matematisk modell for problemet

Vi starter som vanlig med å definere datastørrelsene.

Data:

Vi har kun prosesseringstidene:

$$p_{A1} = 3 \quad \text{prosesstid for ordre A ved maskin 1} \quad (7.26)$$

$$p_{A2} = 5 \quad \text{prosesstid for ordre A ved maskin 2} \quad (7.27)$$

$$p_{B1} = 3 \quad \text{prosesstid for ordre B ved maskin 1} \quad (7.28)$$

$$p_{B2} = 2 \quad \text{prosesstid for ordre B ved maskin 2} \quad (7.29)$$

Variabler:

Tidspunkt:

Vi skal bestemme *tidspunktene* for når ordrene *starter* på hver maskin.

Vi definerer derfor variablene:

$$x_{A1} = \text{startstid for ordre A ved maskin 1} \quad (7.30)$$

$$x_{A2} = \text{startstid for ordre A ved maskin 2} \quad (7.31)$$

$$x_{B1} = \text{startstid for ordre B ved maskin 1} \quad (7.32)$$

$$x_{B2} = \text{startstid for ordre B ved maskin 2} \quad (7.33)$$

Rekkefølge:

Legg merke til at *enten* starter A før B *eller* så starter B før A, siden maskinene ikke kan prosessere ordrene samtidig. For å modellere denne *enten eller* situasjonen, definerer vi *binærvariablene*

$$U_{AB1} = \text{hvorvidt ordre A starter før B på maskin 1, 1 betyr ja, 0 betyr nei} \quad (7.34)$$

$$U_{BA1} = \text{hvorvidt ordre B starter før A på maskin 1, 1 betyr ja, 0 betyr nei} \quad (7.35)$$

$$U_{AB2} = \text{hvorvidt ordre A starter før B på maskin 2, 1 betyr ja, 0 betyr nei} \quad (7.36)$$

$$U_{BA2} = \text{hvorvidt ordre B starter før A på maskin 2, 1 betyr ja, 0 betyr nei} \quad (7.37)$$

Legg merke til at variablene U_{AB1} og U_{BA1} , sier det *motsatte*. Det er derfor en *logisk* sammenheng mellom dem:

$$U_{AB1} = 1 \quad \text{så må} \quad U_{BA1} = 0 \quad (7.38)$$

$$U_{AB1} = 0 \quad \text{så må} \quad U_{BA1} = 1 \quad (7.39)$$

$$(7.40)$$

$$U_{AB2} = 1 \quad \text{så må} \quad U_{BA2} = 0 \quad (7.41)$$

$$U_{AB2} = 0 \quad \text{så må} \quad U_{BA2} = 1 \quad (7.42)$$

$$(7.43)$$

Disse logiske sammenhengene kan vi enkelt beskrive matematisk som:

$$U_{AB1} = 1 - U_{BA1} \quad (7.44)$$

$$U_{AB2} = 1 - U_{BA2} \quad (7.45)$$

Sjekk gjerne at de logiske sammenhengene er oppfylt, dvs sett inn 1 og 0 i likningene.

Konsekvensen av disse sammenhengene er at variablene U_{BA1} og U_{BA2} kan, overalt i den matematiske modellen hvor de forekommer, byttes utde ekvivalente uttrykkene $1 - U_{AB1}$ og $1 - U_{AB2}$ henholdsvis. Vi *reduserer* dermed problemet til kun variablene U_{AB1} og U_{AB2} .

Målfunksjoner:

Vi skal i oppgaven beskrive *to* forskjellige målfunksjoner:

- *Makespan* - *tidligst mulig ferdig*
- *Total gjennomløpstid*

Makespan:

Vi husker at makespan er definert som *sluttidspunktet* for siste ferdigstilte ordre. Merk at vi har følgende sammenheng mellom sluttidspunkt og startidspunkt for en ordre i på en gitt maskin m :

$$\text{sluttidspunkt} = \text{startidspunkt} + \text{prosesseringsid} \quad (7.46)$$

$$= X_{im} + p_{im} \quad (7.47)$$

Makespan er enkel å forstå, men dessverre litt komplisert å beskrive matematisk. Det kommer av at vi i matematisk modellering, ønsker å bruke *linære uttrykk* når vi kan. Årsaken er at linære uttrykk er en forutsetning for løsningsprogrammer som f.eks. Lingo.

Makespan uttrykkes som det *maksimale* sluttidspunktet *for alle ordrene*. Siden vi har to ordrer, får vi at

$$\text{makespan } Y = \max \left[X_{A2} + p_{A2}, X_{B2} + p_{B2} \right] \quad (7.48)$$

Problemet er at makespan Y , *ikke* er en linær funksjon. For å få modellen på linær form, gjør vi følgende to-steps triks:

- Definerer makespan som en *variabel* Y , som da også blir selve målfunksjonen.
- Innfører *føringene* om at alle sluttidspunkt for **alle** ordrene må være mindre enn eller lik Y .

I vårt tilfelle får vi *to ekstra* føringer, siden vi har to ordrer:

$$X_{A2} + p_{A2} \leq Y \quad (7.49)$$

som er sluttidspunkt for ordre A må være mindre enn makespan.

$$X_{B2} + p_{B2} \leq Y \quad (7.50)$$

sluttidspunkt for ordre B må være mindre enn makespan.

Vi godtar både om dere bruker Y -variabel eller max direkte

Total gjennomløpstid

Total gjennomløpstid er adskillig enklere å uttrykke, siden den er gitt som *summen* av alle ordrenes sluttidspunkt:

$$\text{total gjennomløpstid} = \text{sluttidspunkt ordre } A + \text{sluttidspunkt ordre } B \quad (7.51)$$

$$= X_{A2} + p_{A2} + X_{B2} + p_{B2} \quad (7.52)$$

Føringer:

Vi har *to* logiske føringer i modellen:

- *Presedensføringer*, som sier at alle ordrene må først gå gjennom maskin 1, *deretter* gjennom maskin 2.
- *Ikke-samtidighetsføringer*, som sier at en maskin ikke kan prosessere to ordrer *samtidig*.

Presedensføringer:

Presedensføringene sier, for hver ordre, at

$$\text{sluttidspunkt maskin 1} \leq \text{starttidspunkt maskin 2} \quad (7.53)$$

Vi får dermed to føringer, en for hver ordre:

$$X_{A1} + p_{A1} \leq X_{A2} \quad (7.54)$$

$$X_{B1} + p_{B1} \leq X_{B2} \quad (7.55)$$

Ikke-samtidighetsføringer:

Disse føringerne er *logiske* føringer som kommer av sammenhengen mellom tidspunktvariablene (X-ene) og rekkefølgevariablene (U-ene). Føringerne sier at en maskin ikke kan prosessere to ordrer *samtidig*. Vi har følgende logiske sammenheng for maskin 1:

$$\text{Hvis } U_{AB1} = 1, \text{ så må sluttidspunkt } A \text{ maskin 1 være før starttidspunkt } B \text{ maskin 1} \quad (7.56)$$

$$\text{Hvis } U_{AB1} = 0, \text{ så må sluttidspunkt } B \text{ maskin 1 være før starttidspunkt } A \text{ maskin 1} \quad (7.57)$$

Vi kan beskrive sammenhengene mer matematisk ved å bruke uttrykkene for sluttid og starttid:

$$\text{Hvis } U_{AB1} = 1, \text{ så må } X_{A1} + p_{A1} \leq X_{B1} \quad (7.58)$$

$$\text{Hvis } U_{AB1} = 0, \text{ så må } X_{B1} + p_{B1} \leq X_{A1} \quad (7.59)$$

Nå kommer vi til det punktet som ofte er litt komplisert, nemlig at vi fremdeles ønsker å uttrykke føringerne mer matematisk, slik at programpakker som LINGO etc, forstår føringerne. Vi har *to* likeverdige måter å beskrive disse sammenhengene på :

1) Ikke-linær form:²

Dette er den mest *intuitive* formen å uttrykke disse føringene på . Vi *multipliserer* simpelthen BEGGE sider av likningene med hhv U_{AB1} og $U_{BA1} = 1 - U_{AB1}$:

$$U_{AB1}(X_{A1} + p_{A1}) \leq U_{AB1}X_{B1} \quad \text{aktiv dersom } U_{AB1} = 1, \text{ dvs. A før B} \quad (7.60)$$

$$(1 - U_{AB1})(X_{B1} + p_{B1}) \leq (1 - U_{AB1})X_{A1} \quad \text{aktiv dersom } U_{AB1} = 0, \text{ dvs. B før A} \quad (7.61)$$

La oss sjekke at disse føringene faktisk uttrykker det samme som de mer logiske beskrivelsene over:

$$U_{AB1} = 1 \text{ gir } \begin{cases} X_{A1} + p_{A1} \leq X_{B1} & \text{aktiv føring, dvs A før B} \\ 0 \leq 0 & \text{inaktiv føring} \end{cases} \quad (7.62)$$

$$(7.63)$$

$$U_{AB1} = 0 \text{ gir } \begin{cases} 0 \leq 0 & \text{inaktiv føring} \\ X_{B1} + p_{B1} \leq X_{A1} & \text{aktiv føring, dvs B før A} \end{cases} \quad (7.64)$$

$$(7.65)$$

Det er dermed lett å se at uttrykkene oppfyller de logiske betingelsene. Vi får tilsvarende på maskin 2 (bytter bare 1 med 2):

$$U_{AB2}(X_{A2} + p_{A2}) \leq U_{AB2}X_{B2} \quad \text{aktiv dersom } U_{AB2} = 1, \text{ dvs. A før B} \quad (7.66)$$

$$(1 - U_{AB2})(X_{B2} + p_{B2}) \leq (1 - U_{AB2})X_{A2} \quad \text{aktiv dersom } U_{AB2} = 0, \text{ dvs B før A} \quad (7.67)$$

²Denne formen er en såkalt *ikke-lineær* form, dvs vi *multipliserer* to variabler med hverandre (U-ene og X-ene). Selv om LINGO taklet denne formen helt fint, er det generelt et problem for løsningspakker som LINGO. Man ønsker gjerne føringene på lineær form (alternativet under). I dette faget godkjenner vi BEGGE alternativene (men velg en av dem).

2) Lineær form (Stor-M)

Vi bruker et triks kalt stor-M notasjon, hvor hensikten er å få føringene på såkalt *lineær form* (noe som er svært hensiktsmessig for programpakker som LINGO etc, men kanskje ikke like intuitiv som alternativ 1 over). Anta derfor at M er et stort positivt tall. Da kan vi beskrive de logiske føringene over ved:

$$X_{A1} + p_{A1} \leq X_{B1} + M(1 - U_{AB1}) \quad \text{aktiv dersom } U_{AB1} = 1, \text{ dvs A før B} \quad (7.68)$$

$$X_{B1} + p_{B1} \leq X_{A1} + MU_{AB1} \quad \text{aktiv dersom } U_{AB1} = 0, \text{ dvs B før A} \quad (7.69)$$

$$(7.70)$$

La oss sjekke at disse føringene faktisk uttrykker det samme som de mer logiske beskrivelsene over:

$$U_{AB1} = 1 \text{ gir } \begin{cases} X_{A1} + p_{A1} \leq X_{B1} + M(1 - 1) = X_{B1} & \text{aktiv føring, dvs A før B} \\ X_{B1} + p_{B1} \leq X_{A1} + M(1 - 0) = X_{A1} + M & \text{inaktiv føring} \end{cases} \quad (7.71)$$

$$U_{AB1} = 0 \text{ gir } \begin{cases} X_{A1} + p_{A1} \leq X_{B1} + M(1 - 0) = X_{B1} + M & \text{inaktiv føring} \\ X_{B1} + p_{B1} \leq X_{A1} + M \cdot 0 = X_{A1} & \text{aktiv føring, dvs B før A} \end{cases} \quad (7.72)$$

Legg merke til at de *inaktive* føringene blir faktisk inaktive siden vi kan velge M så stor at ulikheten [alltid](#) er oppfylt. Dette kan vi oppnå for eksempel ved å velge M lik *summen* av alle prosesserings-tidene. I vårt eksempel kan vi derfor velge $M=3+5+4+2=14$

Tilsvarende får vi for maskin 2:

$$X_{A2} + p_{A2} \leq X_{B2} + M(1 - U_{AB2}) \quad \text{aktiv dersom } U_{AB2} = 1, \text{ dvs. A før B} \quad (7.73)$$

$$X_{B2} + p_{B2} \leq X_{A2} + MU_{AB2} \quad \text{aktiv dersom } U_{AB2} = 0, \text{ dvs. B før A} \quad (7.74)$$

$$(7.75)$$

LINGO program: MAKESPAN

Min Y

$$XA2 + pA2 \leq Y;$$

$$XB2 + pB2 \leq Y;$$

$$XA1 + pA1 \leq XA2;$$

$$XB1 + pB1 \leq XB2;$$

$$UAB1 * (XA1 + pA1) \leq UAB1 * XB1;$$

$$(1 - UAB1) * (XB1 + pB1) \leq (1 - UAB1) * XA1;$$

$$UAB2 * (XA2 + pA2) \leq UAB2 * XB2;$$

$$(1 - UAB2) * (XB2 + pB2) \leq (1 - UAB2) * XA2;$$

$$@Bin(UAB1);$$

$$@Bin(UAB2);$$

LINGO program: Total gjennomløpstid

Min $XA2 + pA2 + XB2 + pB2$

$$XA1 + pA1 \leq XA2;$$

$$XB1 + pB1 \leq XB2;$$

$$UAB1 * (XA1 + pA1) \leq UAB1 * XB1;$$

$$(1 - UAB1) * (XB1 + pB1) \leq (1 - UAB1) * XA1;$$

$$UAB2 * (XA2 + pA2) \leq UAB2 * XB2;$$

$$(1 - UAB2) * (XB2 + pB2) \leq (1 - UAB2) * XA2;$$

$$@Bin(UAB1);$$

$$@Bin(UAB2);$$

bi) Med en ny ordre C øker problemets kompleksitet. Vi har 6 mulige optimale planer:

$$ABC = \text{Først A, så B så C} \quad (7.76)$$

$$ACB = \text{Først A, så C så B} \quad (7.77)$$

$$BAC = \text{Først B, så A så C} \quad (7.78)$$

$$BCA = \text{Først B, så C så A} \quad (7.79)$$

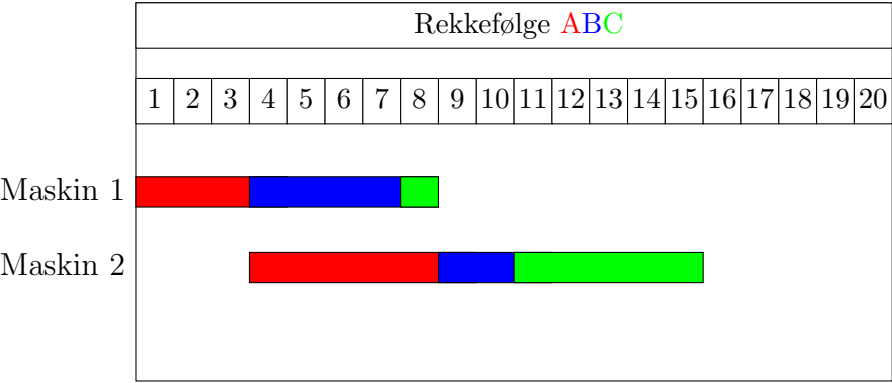
$$CAB = \text{Først C, så A så B} \quad (7.80)$$

$$CBA = \text{Først C, så B så A} \quad (7.81)$$

$$(7.82)$$

Vi må dermed lage 6 Gantt-diagrammer for å bestemme hvilke av dem som er optimale.

ABC:



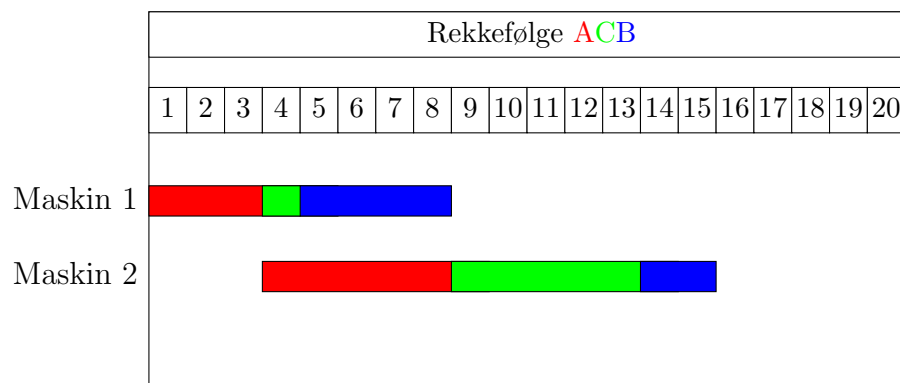
Figur 7.14: Rekkefølge ABC.

$$\text{makespan} = \text{sluttid } C = \underline{\underline{15 \text{ timer}}} \tag{7.83}$$

$$\begin{aligned} \text{total gjennomløpstid} &= \text{sluttid } A + \text{sluttid } B + \text{sluttid } C \\ &= 8 + 10 + 15 = \underline{\underline{33 \text{ timer}}} \end{aligned} \tag{7.84}$$

$$\underline{\underline{\text{gjennomsnittlig gjennomløpstid}}} = \frac{\text{total gjennomløpstid}}{3} = \frac{33}{3} = \underline{\underline{11 \text{ timer}}} \tag{7.85}$$

ACB:



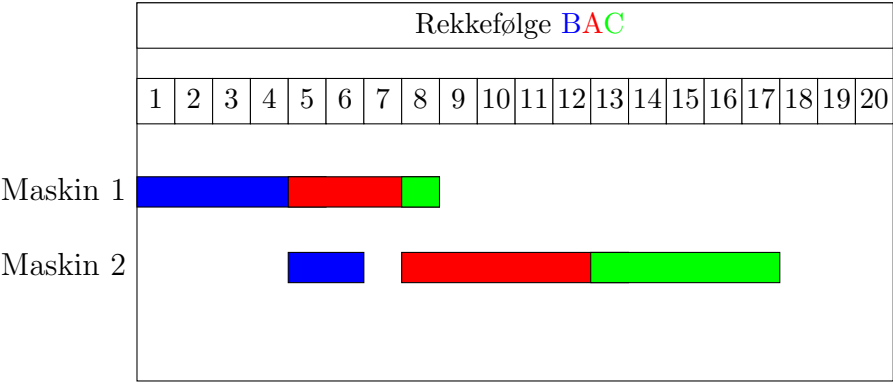
Figur 7.15: Rekkefølge ACB.

$$\underline{\underline{\text{makespan}}} = \text{sluttid } B = \underline{\underline{15 \text{ timer}}} \quad (7.86)$$

$$\begin{aligned} \underline{\underline{\text{total gjennomløpstid}}} &= \text{sluttid } A + \text{sluttid } B + \text{sluttid } C \\ &= 8 + 13 + 15 = \underline{\underline{35 \text{ timer}}} \end{aligned} \quad (7.87)$$

$$\underline{\underline{\text{gjennomsnittlig gjennomløpstid}}} = \frac{\text{total gjennomløpstid}}{3} = \frac{35}{3} = \underline{\underline{11.666 \text{ timer}}} \quad (7.88)$$

BAC:



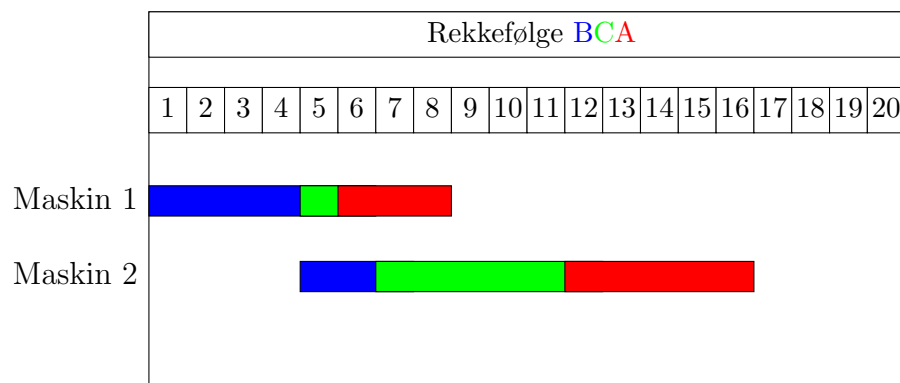
Figur 7.16: Rekkefølge BAC.

$$\underline{\text{makespan}} = \text{sluttid } C = \underline{17 \text{ timer}} \tag{7.89}$$

$$\begin{aligned} \underline{\text{total gjennomløpstid}} &= \text{sluttid } A + \text{sluttid } B + \text{sluttid } C \\ &= 6 + 12 + 17 = \underline{35 \text{ timer}} \end{aligned} \tag{7.90}$$

$$\underline{\text{gjennomsnittlig gjennomløpstid}} = \frac{\text{total gjennomløpstid}}{3} = \frac{35}{3} = \underline{11.666 \text{ timer}} \tag{7.91}$$

BCA:



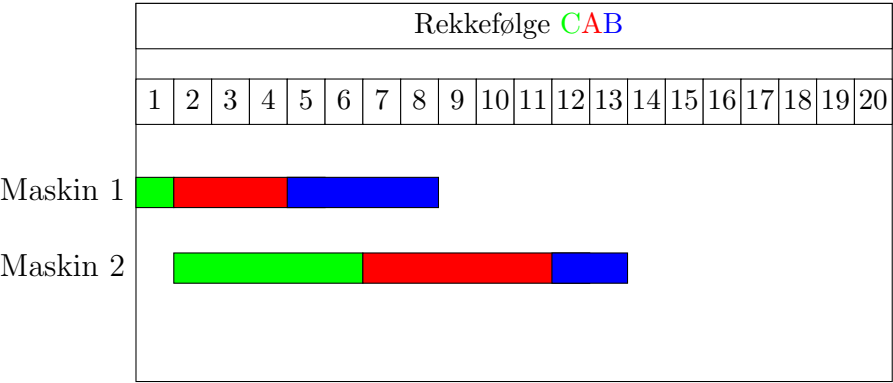
Figur 7.17: Rekkefølge BCA.

$$\underline{\underline{\text{makespan}}} = \text{sluttid } \textcolor{red}{A} = \underline{\underline{16 \text{ timer}}} \quad (7.92)$$

$$\begin{aligned} \underline{\underline{\text{total gjennomløpstid}}} &= \text{sluttid } \textcolor{red}{A} + \text{sluttid } \textcolor{blue}{B} + \text{sluttid } \textcolor{green}{C} \\ &= 6 + 11 + 16 = \underline{\underline{33 \text{ timer}}} \end{aligned} \quad (7.93)$$

$$\underline{\underline{\text{gjennomsnittlig gjennomløpstid}}} = \frac{\text{total gjennomløpstid}}{3} = \frac{33}{3} = \underline{\underline{11 \text{ timer}}} \quad (7.94)$$

CAB:



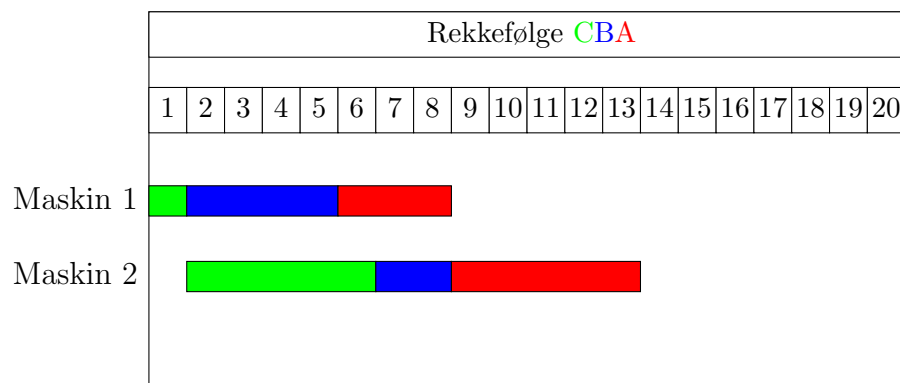
Figur 7.18: Rekkefølge CAB.

$$\text{makespan} = \text{sluttid } B = \underline{\underline{13 \text{ timer}}} \tag{7.95}$$

$$\begin{aligned} \text{total gjennomløpstid} &= \text{sluttid } A + \text{sluttid } B + \text{sluttid } C \\ &= 6 + 11 + 13 = \underline{\underline{30 \text{ timer}}} \end{aligned} \tag{7.96}$$

$$\underline{\underline{\text{gjennomsnittlig gjennomløpstid}}} = \frac{\text{total gjennomløpstid}}{3} = \frac{30}{3} = \underline{\underline{10 \text{ timer}}} \tag{7.97}$$

CBA:



Figur 7.19: Rekkefølge CBA.

$$\underline{\underline{\text{makespan}}} = \text{sluttid } A = \underline{\underline{13 \text{ timer}}} \quad (7.98)$$

$$\begin{aligned} \underline{\underline{\text{total gjennomløpstid}}} &= \text{sluttid } A + \text{sluttid } B + \text{sluttid } C \\ &= 6 + 8 + 13 = \underline{\underline{27 \text{ timer}}} \end{aligned} \quad (7.99)$$

$$\underline{\underline{\text{gjennomsnittlig gjennomløpstid}}} = \frac{\text{total gjennomløpstid}}{3} = \frac{27}{3} = \underline{\underline{9 \text{ timer}}} \quad (7.100)$$

Konklusjon:

Minimalt makespan: 13 timer ved rekkefølgene CAB og CBA

Minimal gjennomløpstid: 27 timer ved rekkefølgen CBA

b ii) Vi har fått en ny ordre C, som må inkluderes i modellen:

Nye Data:

$$p_{C1} = 1 \quad \text{prosesstid for ordre C ved maskin 1} \quad (7.101)$$

$$p_{C2} = 5 \quad \text{prosesstid for ordre C ved maskin 2} \quad (7.102)$$

Nye Variabler:

Vi får to nye startstidspunkt som må bestemmes:

$$x_{C1} = \text{startstid for ordre C ved maskin 1} \quad (7.103)$$

$$x_{C2} = \text{startstid for ordre C ved maskin 2} \quad (7.104)$$

Når det gjelder hvilken *rekkefølge* ordrene skal prosesseres, må vi definere rekkefølgen til ordre C *både* i forhold til A og B:

$$U_{AC1} = \text{hvorvidt ordre A starter før C på maskin 1, 1 betyr ja} \quad (7.105)$$

$$U_{BC1} = \text{hvorvidt ordre B starter før C på maskin 1, 1 betyr ja} \quad (7.106)$$

$$U_{AC2} = \text{hvorvidt ordre A starter før C på maskin 2, 1 betyr ja} \quad (7.107)$$

$$U_{BC2} = \text{hvorvidt ordre B starter før C på maskin 2, 1 betyr ja} \quad (7.108)$$

Utvidede målfunksjoner:

Begge målfunksjonene må inkludere den nye ordren:

Makespan:

Vi husker vi definerer makespan Y som en *variable*:

$$\text{makespan } Y = \max \left[X_{A2} + p_{A2} , X_{B2} + p_{B2} , X_{C2} + p_{C2} \right] \quad (7.109)$$

I tillegg må vi legge til føringen:

$$X_{C2} + p_{C2} \leq Y \quad (7.110)$$

som er sluttidspunkt for ordre C må være mindre enn makespan

Total gjennomløpstid:

Vi må inkludere sluttidspunktet til ordre C i den totale gjennomløpstiden:

$$\text{total gjennomløpstid} = X_{A2} + p_{A2} + X_{B2} + p_{B2} + X_{C2} + p_{C2} \quad (7.111)$$

Nye Føringar:

Ny presedensføring:

Vi får en ny presedensføring for ordre C (maskin 1 før maskin 2):

$$X_{C1} + p_{C1} \leq X_{C2} \quad (7.112)$$

Nye ikke-samtidighetsføringer:

Vi får fire nye sammenhenger mellom tidspunkt og rekkefølge.

Vi husker vi har to alternativ:

1) Ikke-linær form:

For maskin 1:

$$U_{AC1}(X_{A1} + p_{A1}) \leq U_{AC1}X_{C1} \quad \text{aktiv dersom } U_{AC1} = 1, \text{ dvs. A før C} \quad (7.113)$$

$$(1 - U_{AC1})(X_{C1} + p_{C1}) \leq (1 - U_{AC1})X_{A1} \quad \text{aktiv dersom } U_{AC1} = 0, \text{ dvs C før A} \quad (7.114)$$

$$U_{BC1}(X_{B1} + p_{B1}) \leq U_{BC1}X_{C1} \quad \text{aktiv dersom } U_{BC1} = 1, \text{ dvs. B før C} \quad (7.115)$$

$$(1 - U_{BC1})(X_{C1} + p_{C1}) \leq (1 - U_{BC1})X_{B1} \quad \text{aktiv dersom } U_{BC1} = 0, \text{ dvs C før B} \quad (7.116)$$

For maskin 2:

$$U_{AC2}(X_{A2} + p_{A2}) \leq U_{AC2}X_{C2} \quad \text{aktiv dersom } U_{AC1} = 1, \text{ dvs. A før C} \quad (7.117)$$

$$(1 - U_{AC2})(X_{C2} + p_{C2}) \leq (1 - U_{AC2})X_{A2} \quad \text{aktiv dersom } U_{AC2} = 0, \text{ dvs. C før A} \quad (7.118)$$

$$U_{BC2}(X_{B2} + p_{B2}) \leq U_{BC2}X_{C2} \quad \text{aktiv dersom } U_{BC2} = 1, \text{ dvs. B før C} \quad (7.119)$$

$$(1 - U_{BC2})(X_{C2} + p_{C2}) \leq (1 - U_{BC2})X_{B2} \quad \text{aktiv dersom } U_{BC2} = 0, \text{ dvs. C før B} \quad (7.120)$$

2) Linær form (Stor M):

For maskin 1:

$$X_{A1} + p_{A1} \leq X_{C1} + M(1 - U_{AC1}) \quad \text{aktiv dersom } U_{AC1} = 1, \text{ dvs. A før C} \quad (7.121)$$

$$X_{C1} + p_{C1} \leq X_{A1} + MU_{AC1} \quad \text{aktiv dersom } U_{AC1} = 0, \text{ dvs. C før A} \quad (7.122)$$

$$X_{B1} + p_{B1} \leq X_{C1} + M(1 - U_{BC1}) \quad \text{aktiv dersom } U_{BC1} = 1, \text{ dvs. B før C} \quad (7.123)$$

$$X_{C1} + p_{C1} \leq X_{B1} + MU_{BC1} \quad \text{aktiv dersom } U_{BC1} = 0, \text{ dvs. C før B} \quad (7.124)$$

For maskin 2:

$$X_{A2} + p_{A2} \leq X_{C2} + M(1 - U_{AC2}) \quad \text{aktiv dersom } U_{AC2} = 1, \text{ dvs. A før C} \quad (7.125)$$

$$X_{C2} + p_{C2} \leq X_{A2} + MU_{AC2} \quad \text{aktiv dersom } U_{AC2} = 0, \text{ dvs. C før A} \quad (7.126)$$

$$X_{B2} + p_{B2} \leq X_{C2} + M(1 - U_{BC2}) \quad \text{aktiv dersom } U_{BC2} = 1, \text{ dvs. B før C} \quad (7.127)$$

$$X_{C2} + p_{C2} \leq X_{B2} + MU_{BC2} \quad \text{aktiv dersom } U_{BC2} = 0, \text{ dvs. C før B} \quad (7.128)$$

Husk: hvilket alternativ dere bruker er opp til dere, bare dere velger minst en av dem!

LINGO program: MAKESPAN

Min Y

$$XA2 + pA2 \leq Y;$$

$$XB2 + pB2 \leq Y;$$

$$XC2 + pC2 \leq Y;$$

$$XA1 + pA1 \leq XA2;$$

$$XB1 + pB1 \leq XB2;$$

$$XC1 + pC1 \leq XC2;$$

$$UAB1 * (XA1 + pA1) \leq UAB1 * XB1;$$

$$(1 - UAB1) * (XB1 + pB1) \leq (1 - UAB1) * XA1;$$

$$UAC1 * (XA1 + pA1) \leq UAC1 * XC1;$$

$$(1 - UAC1) * (XC1 + pC1) \leq (1 - UAC1) * XA1;$$

$$UBC1 * (XB1 + pB1) \leq UBC1 * XC1;$$

$$(1 - UBC1) * (XC1 + pC1) \leq (1 - UBC1) * XB1;$$

$$UAB2 * (XA2 + pA2) \leq UAB2 * XB2;$$

$$(1 - UAB2) * (XB2 + pB2) \leq (1 - UAB2) * XA2;$$

$$UAC2 * (XA2 + pA2) \leq UAC2 * XC2;$$

$$(1 - UAC2) * (XC2 + pC2) \leq (1 - UAC2) * XA2;$$

$$UBC2 * (XB2 + pB2) \leq UBC2 * XC2;$$

$$(1 - UBC2) * (XC2 + pC2) \leq (1 - UBC2) * XB2;$$

$$@Bin(UAB1);$$

$$@Bin(UAB2);$$

$$@Bin(UAC1);$$

$$@Bin(UAC2);$$

$$@Bin(UBC1);$$

$$@Bin(UBC2);$$

LINGO program: Total gjennomløpstid

$$\text{Min } XA2 + pA2 + XB2 + pB2 + XC2 + pC2$$

$$XA1 + pA1 \leq XA2;$$

$$XB1 + pB1 \leq XB2;$$

$$XC1 + pC1 \leq XC2;$$

$$UAB1 * (XA1 + pA1) \leq UAB1 * XB1;$$

$$(1 - UAB1) * (XB1 + pB1) \leq (1 - UAB1) * XA1;$$

$$UAC1 * (XA1 + pA1) \leq UAC1 * XC1;$$

$$(1 - UAC1) * (XC1 + pC1) \leq (1 - UAC1) * XA1;$$

$$UBC1 * (XB1 + pB1) \leq UBC1 * XC1;$$

$$(1 - UBC1) * (XC1 + pC1) \leq (1 - UBC1) * XB1;$$

$$UAB2 * (XA2 + pA2) \leq UAB2 * XB2;$$

$$(1 - UAB2) * (XB2 + pB2) \leq (1 - UAB2) * XA2;$$

$$UAC2 * (XA2 + pA2) \leq UAC2 * XC2;$$

$$(1 - UAC2) * (XC2 + pC2) \leq (1 - UAC2) * XA2;$$

$$UBC2 * (XB2 + pB2) \leq UBC2 * XC2;$$

$$(1 - UBC2) * (XC2 + pC2) \leq (1 - UBC2) * XB2;$$

$$@Bin(UAB1);$$

$$@Bin(UAB2);$$

$$@Bin(UAC1);$$

$$@Bin(UAC2);$$

$$@Bin(UBC1);$$

$$@Bin(UBC2);$$