

Inhoudsopgave

Deel I: Facilities Design	4
1. Inleiding	4
A. Wat is Facilities Design?	4
B. Types layouts vs process flexibility/product variety	4
1. Dokstysteem	4
2. Functionele Lay-out	4
3. Flowproductie	5
4. Celproductie	5
5. Continue productie	5
6. Hybride layouts	5
C. KOOP	6
D. Belang van Layout beslissingen	6
2. Functionele Layout Planning (= FLP)	6
A. DATA	6
i) Functionele Layout = Jobshop	6
ii) Benodigde data = FLOW DATA	6
B. METHODEN	6
i) Kwantitatieve Methoden	7
I) Lineair toewijzingsprobleem	7
Oplossingsmethode: Hongaarse methode	7
II) Kwadratisch Toewijzingsprobleem	8
Oplossingsmethode 1 (EXACTE METHODE) : Branch and Bound	8
Hoe berekenen we de Lowerbound?	8
Hoe berekenen we de Upperbound?	8
Oplossingsmethode 2 (HEURISTIEK): CRAFT	9
i) Kwalitatieve Methoden	9
I) Grafentheoretische methode	9
3. Group Technology	9
A. PPI matrix	9
B. Clusteringalgoritme methodes	9
i) R&CM algoritme	9
ii) ROC algoritme	10
iii) SC algoritme	10
iiii) MP Approach (kunnen opstellen en oplossing interpreteren)	10
4. Assembly Line Balancing	11
A. Inleiding	11
i) Single-model lijnen	11
B. Branch and Bound voor single model lijnen	11

Hoe bepaal je de knooppunten?.....	11
Gedomineerde knooppunten	11
Benedengrens per knooppunt	11
Vertakkingsregels.....	12
C. Heuristieken voor Single model lijnen	12
i) Ranked Positional Weight	12
ii) Hoffmann Heuristiek	12
I) Forward Balance	12
II) Reverse Balance	12
D. U-vormige lijnen	12
E. Mixed Model Line Balancing	12
i) Hoe balanceren?	12
ii) Minimale leegloop minimaliseren	12
I) Voordelen	12
II) Nadelen.....	12
iii) RPW	12
iiii) Repetitieve cycli	12
iiiii) Schendingen van takttijd wegwerken?	13
5. Warehousing.....	13
A. Warehouse functions and flows	13
B. Order picking systems.....	13
i) Picker-to-parts	13
I) Low level picking	13
II) High level picking	13
ii) Parts-to-picker.....	13
C. Design of order picking systems.....	13
i) How to improve Setup, pick and order time.....	14
I) Paper picklists	14
II) Pick-by-voice	14
III) Pick-by-light	14
IV) Pick by HMD.....	14
ii) How to improve the travel time?	14
I) Layout Design.....	14
II) Storage assignment.....	14
1. Random	14
2. Dedicated	14
3. Full-turnover storage	14
4. Class-based storage.....	14
III) Batching	14

1. Pick-by-Order = discrete picking	14
2. Pick-by-Article = batch picking	14
IV) Zoning	14
1. Pick and pass	14
2. Parallel picking	14
V) Routing.....	14
1. S-shape	14
2. Return.....	14
3. Mid-point.....	15
4. Largest Gap	15

Organisatie van het Productieproces

Deel I: Facilities Design

1. Inleiding

A. Wat is Facilities Design?

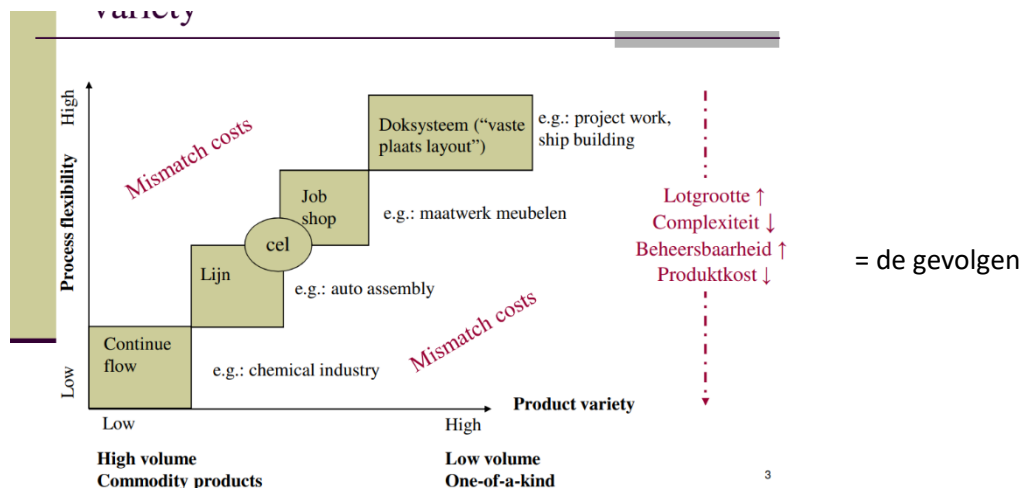
~~Structural Design~~

Layout Design → Layout van de productieruimten, ondersteunende ruimten en personeelsruimten

~~Handling System Design~~

B. Types layouts vs process flexibility/product variety

90% van de huidige systemen:



- Cel is een overgangsvorm tussen lijnstructuur en jobshop.
- Mismatch costs willen we NIET.
- Dit zijn de 5 basistypes, gelijkaardige types processen vertonen gelijkaardige beheersproblemen.
- Er is een link met de levenscyclus van het product (bv starten als startup in jobshop en op termijn naar een continue flow door de groei) → **productie layout moet mee evolueren**

1. Dokstelsysteem

= Product blijft ter plaatse, bedrijfsmiddelen bewegen. (extreme vorm)

'Dok' maakt het product volledig gereed

Voordelen	Nadelen
Product beweegt niet	Beweging van personeel en uitrusting, risico overschrijding deadlines
	Duplicatie van uitrusting (→ lage bezettingsgraad)
Gekwalificeerd personeel + taakverrijking: dokstelsysteem laat toe dat individuen/teams "de hele job" uitvoeren	Algemeen toezicht vereist
Zeer flexibel: laat veranderingen toe in productontwerp/productmix	

2. Functionele Lay-out

= jobshop = Process Layout

- Machines staan gegroepeerd in afdelingen volgens het type bewerking dat ze uitvoeren.
- Elk product volgt een eigen route → flexibel
- Kleine herhalingsgraad van de vraag met lage volumes.

Voordelen	Nadelen
Gespecialiseerde kennis, vakspecialisatie	Personeel niet inzetbaar buiten eigen departement
Zeer hoge flexibiiteit in productontwerp/product mix: universele middelen, flexibele routings	

Voordelen	Nadelen
Machines kunnen op meerdere stappen in productieproces gebruikt worden: hoge bezettingsgraad, lage investeringskost (→ flow shop)	Sterk variërende routings, chaotische flows
	Productieplanning en productiecontrole = moeilijk: - Grote variabiliteit in procestijden - Omsteltijden
	Grote hoeveelheid work-in-process (WIP) op de vloer Vereist werkkapitaal + ruimte !!
	Doorlooptijd is vaak een veelvoud van bewerkingstijd van een order
	Expediting, rush orders

3. Flowproductie

= Product layout

- Machines staan gegroepeerd in lijnen, gericht op de productie van een bepaald type product
- Alle producten op die lijn volgen dezelfde bewerkingsvolgorde.
- 2 types:

Gebonden Lijn	Ongebonden Lijn
Geen WIP of buffers tussen stations	Buffers tussen de stations = WIP
Eenzelfde ritme per station	Stations werken onafhankelijk van elkaar
<i>Continu lopende band</i> = proces gaat gewoon door, operator beweegt mee	
<i>Intermitterende lijn</i> = geen bewerkingen tijdens verplaatsing	

Voordelen	Nadelen
De layout stemt overeen met de volgorde van de bewerkingen • ordelijke en logische flow • beperkte material handling Lijnbalancerings: • Bewerkingen per station worden zodanig gegroepeerd dat elk station de taktijd haalt • Geen opbouw van WIP tussen de stations (producten en materiaal voortdurend in beweging) • Lage doorlooptijd (minimale wachttijden) Goede beheersbaarheid (planning, controle) Mogelijkheid voor automatisering (processen, material handling)	Duplicatie van machinetypes: als identieke machines op meerdere plaatsen in het proces gebruikt worden • in dat geval: lage bezettingsgraad)

Voordelen	Nadelen
Beperkte vaardigheden vereist van arbeiders: training = kort, eenvoudig, goedkoop	Geringere werkvoltoening, vervreemding werkers. Geringe flexibiliteit • Breakdown op bepaalde machine → shutdown lijn (vooral bij gebonden lijn). • Lage ombouwflexibiliteit: verandering in productmix → verandering lijnontwerp
Lage kost per unit • Gerichtheid op specifiek type product (beperkte productmix) • Hoge volumes	Hoge investeringskosten, dus investering is enkel gerechtvaardigd voor producten met voldoende grote vraag (massaproductie).

4. Celproductie

= Group Technology Based layout

DOEL= combinatie van efficiëntie en flexibiliteit

- Machines die bepaalde productfamilie maken, worden gegroepeerd in een cel.
- Idealiter: geen materiaal handling tussen de cellen
- Binnen de cel: hogere efficiëntie door minder omstellingen en minder materiaal handling
- Tussenstap van jobshop → flowproductie

GT cel = minifabriek

- Functionele layout binnen een cel
- Elk lid van de familie volgt een eigen pad binnenin de cel
- Nuttig: grote variëteit + klein volume + groot aantal verschillende routings

GT Flow Line = JIT cel

- Process layout binnen een cel
- Ongebonden lijnstructuur binnen de cel → Buffers + bewerkingstijd afhankelijk/product
- U-shape layout (operatoren helpen elkaar + kortere afstanden)

Voordelen	Nadelen
Vereenvoudigde goederenstroom tov process layout: betere beheersbaarheid, eenvoudige productopvolging Kortere doorlooptijden en minder WIP dan bij process layout	Machines worden apart geplaatst in cel en behandelen enkel de productfamilie: lagere bezettingsgraad dan process layout
Teamwerk, verhoogde werkvoltoening en verantwoordelijkheid arbeiders	
Hogere flexibiliteit dan product layout, hogere efficiëntie dan process layout	

5. Continue productie

= extreem geval van lijnproductie

- Grote herhalingsgraad + tussenvoorraden Onmogelijk
- Voor-/nadelen nog meer uitgesproken dan bij een lijnproductie

6. Hybride layouts

Binnen eenzelfde bedrijf kunnen meerdere layout-types voorkomen.

C. KOOP

= Klantenorder-onkoppelpunt

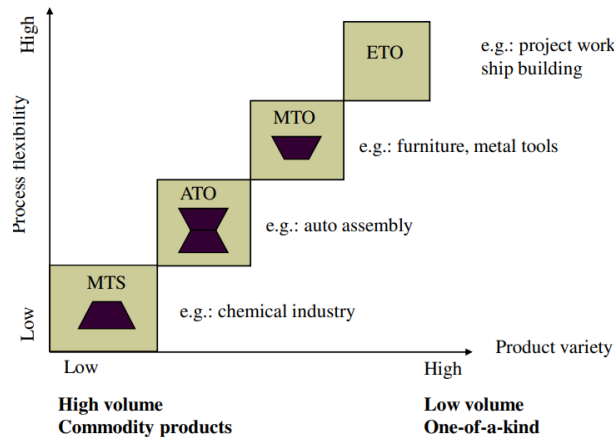
Vanaf dat punt wordt de fabricage aangestuurd door een reële klantenorder.

Voor dit punt → PUSH

Na dit punt → PULL



Verskillende klassen volgens KOOP:



ETO = Engineer To Order (alles is PULL)

MTO = Make To Order

ATO= Assemble To Order

MTS= Make To Stock (alles is PUSH)

D. Belang van Layout beslissingen

- Heeft een impact op performantie system
Doorlooptijd, WIP, Productkost, leveringsbetrouwbaarheid....
- Moeilijk om van typologie af te wijken
- Grote investeringen + tijd
- Nu verandert de layout vaker, door snellere veranderingen in productmix

2. Functionele Layout Planning (= FLP)

A. DATA

i) Functionele Layout = Jobshop

- Kleine herhalingsgraad
- Machines gegroepeerd volgens type bewerking
- Product volgt eigen route doorheen de fabriek
➔ Hoe kunnen we de optimale layout bepalen?

ii) Benodigde data = FLOW DATA

Flow tussen de departementen is een van de belangrijkste factoren in het bepalen van de layout.

Kwantitatieve Flow Data	Kwalitatieve Flow Data
= Meten van fysische flow tussen facility pairs	= closeness ratings (drukken wenselijkheid uit dat 2 faciliteiten naast elkaar gelegen zijn) → geven rangordeschaal weer
→ Aantal strips per tijdseenheid	Codes voor relationship
From-to matrix	N facilities → $N(N-1)/2$ relationships
Trip between matrix (symmetrisch) = optellen symmetrische cellen van een from-to matrix	Relationship Chart= REL chart

From-to matrix opstellen:

1. Part-routing matrix
2. Volume per tijdseenheid
3. Aantal eenheden per lading

B. METHODEN

DOEL = Opstellen van een blokkenschema = plattegrond met aanduiding van afdelingen

Kwantitatieve methoden	Kwalitatieve methoden
Lineair toewijzingsprobleem (Hongaarse Methode)	Grafentheorie
Kwadratisch toewijzingsprobleem (Branch & Bound + CRAFT)	

i) Kwantitatieve Methoden

Optimaliteitsmaatstaf = totale kosten voor material handling

Optimale layout = diegene dat de material handling kosten minimaliseert

Layout x Afstand x Kost

LAYOUT = Kwantitatieve flow data tussen facilities (bv. Aantal strips per dag)

KOST = Kost voor verplaatsing per trip per eenheid per afstand (Zetten we altijd gelijk aan 1)

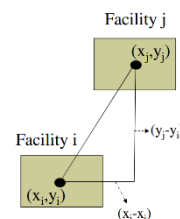
AFSTAND = Afstand tussen locaties van facilities (Gebeurt tussen 2 zwaartepunten van 2 facilities)

Euclidische afstand:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Rechthoekige afstand:

$$d_{ij} = |x_i - x_j| + |y_i - y_j|$$



l) Lineair toewijzingsprobleem

- Nieuwe facilities toewijzen aan de nog beschikbare locaties

- Som van de vaste toewijzingskosten moeten geminimaliseerd worden

Vaste toewijzingskost = kost voor materiaalbeweging tussen nieuw geplaatste facilities en reeds bestaande

- Zinvol als er geen interacties zijn tussen de nieuw te plaatsen facilities onderling (deze kosten worden genegeerd)

NOTATIE:

m = aantal te plaatsen objecten

n = mogelijke locaties

a_{ij} = vaste kost voor het toewijzen van object i aan locatie j

x_{ij} = beslissingsvariable: object i toekennen aan locatie j = 1 of niet = 0

Voor een feasible oplossing $\rightarrow n \geq m$

$$\begin{aligned}
 &\min \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} \quad \rightarrow \text{Minimaliseer vaste toewijzingskosten} \\
 &\text{subject to} \\
 &\sum_{i=1}^m x_{ij} \leq 1 \quad j = 1, 2, \dots, n \quad \rightarrow \text{Elke locatie wordt aan maximum 1 object toegewezen} \\
 &\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, m \quad \rightarrow \text{Elk object MOET aan precies 1 locatie worden toegewezen} \\
 &x_{ij} = \begin{cases} 1 & \text{als facility } i \text{ wordt toegewezen aan locatie } j \\ 0 & \text{in het andere geval} \end{cases}
 \end{aligned}$$

Oplossingsmethode: **Hongaarse methode**

(Voorwaarde: m = n, bij meer locaties dan objecten kan je dummyobjecten toevoegen)

Stap 1: Trek het kleinste getal in elke rij van de a_{ij} matrix af van alle getallen in die rij. Trek dan het kleinste element in elke kolom van de bekomen matrix af van alle getallen in die kolom.

Stap 2: Trek het kleinste aantal lijnen (horizontaal, verticaal, of beide) die alle nullen afdekken. Als n lijnen nodig zijn, is de optimale toewijzing gevonden. Als minder dan n lijnen nodig zijn, ga naar stap 3. Aantal lijnen = dimensie

Stap 3: Vind het kleinste element (niet gelijk aan 0) in de matrix, dat NIET door een lijn afgedekt wordt. Trek dit element af van alle niet-afgedekte elementen in de matrix, en tel het op bij alle elementen die door 2 lijnen worden afgedekt. Ga terug naar stap 2.

(Kunnen toepassen)

II) Kwadratisch Toewijzingsprobleem

- Toewijzen van m faciliteiten aan n locaties, zodat de som van de vaste toewijzingskosten en de interactiekosten word geminimaliseerd
- C_{ijkl} = kost die voortvloeit uit het gelijktijdig toewijzen van i aan j en k aan l
 = materiaalbehandelingskost
 = $f_{ik} q_{ik} d_{jl}$
 D_{jl} = afstand tussen locatie j en l (afstandenmatrix)
 F_{ik} = aantal eenheidsladingen getransporteerd tussen i en k (trip-between matrix)
 Q_{ik} = matriaalbehandelingskost voor transport van 1 lading over 1 eenheid afstand tussen i en k

$$\min \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} + \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^m \sum_{l=1}^n c_{ijkl} x_{ij} x_{kl}$$

onderworpen aan

$$\sum_{i=1}^m x_{ij} \leq 1 \quad j=1,2,\dots,n \quad \longrightarrow \quad \text{Aan elke locatie wordt maximum 1 object toegewezen}$$

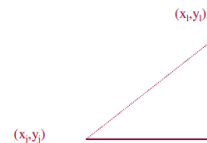
$$\sum_{j=1}^n x_{ij} = 1 \quad i=1,2,\dots,m \quad \longrightarrow \quad \text{Elk object wordt aan precies 1 locatie toegewezen}$$

$$x_{ij} = \begin{cases} 1 & \text{als facility } i \text{ wordt toegewezen aan lokatie } j \\ 0 & \text{in het andere geval} \end{cases}$$

Inputdata: $a_{ij} \quad \forall i, j$

$$c_{ijkl} = f_{ik} q_{ik} d_{jl} \quad \forall i, j, k, l$$

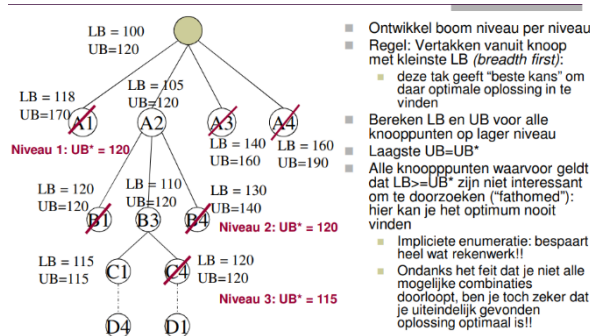
$$d_{jl} = |x_j - x_l| + |y_j - y_l| \text{ of } \sqrt{(x_j - x_l)^2 + (y_j - y_l)^2} \quad \forall j, l$$



Oplossingsmethode 1 (EXACTE METHODE) : Branch and Bound

(Garandeert wiskundig optimale oplossing)

- Werkt via een zoekboom met knooppunten
- Lowerbound = rooskleurige inschatting
- Upperbound = werkelijke oplossing



Hoe berekenen we de Lowerbound?

1. Interactiekosten tussen reeds geplaatste faciliteiten onderling (exact)
2. Interactiekosten tussen reeds geplaatste faciliteiten en nog te plaatsen faciliteiten (exact)
3. Benedengrens op interactiekosten tussen nog te plaatsen faciliteiten (roze bril)
 → Laatste 2 via lineair toewijzingsprobleem → **Hongaarse Methode**
 Kostenmatrix nodig = $w_{ij} = w'_{ij} + w''_{ij}$

Hoe berekenen we de Upperbound?

= Werkelijke material handling kosten die samenhangen met LB-oplossing
 = $F_{jk} d_{jl} / 2$

Oplossingsmethode 2 (HEURISTIEK): CRAFT

= Iteratieve verbeteringsprocedure

- We moeten de coördinaten van de zwaartepunten weten liggen
- From-to matrix kennen we
- Berekening totale kosten:

Flow x Afstand x Kosteneenheid flow per eenheid afstand

Flow = from-to matrix

Afstand = afstand tussen de zwaartepunten

Kosteneenheid = 1

→ Paarsgewijze omwisseling

Zwaartepunten van faciliteiten met dezelfde oppervlakte OF een gemeenschappelijke grens hebben worden omgewisseld

Voor elke omwisseling → estimated material handling cost

Beste omwisseling met laagste kost wordt uitgevoerd

i) Kwalitatieve Methoden

l) Grafentheoretische methode

Graf $G = (N, X)$

N = aantal knooppunten

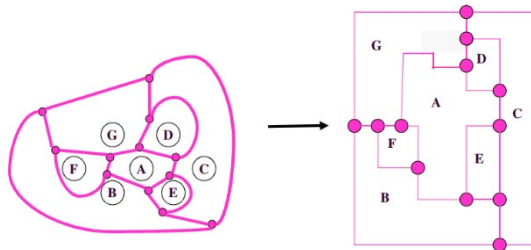
E = aantal edges = bogen

F = aantal faces = aantal oppervlakten (inclusief buitenwereld)

Planaire graf = edges stijden elkaar niet = $n - e + f$

Maximale planaire graf = als er geen enkele edge meer kan toegevoegd worden zonder planariteit op te offeren = $2e = 3f$, $e_{\max} = 3n - 6$

→ Doel: maximaal planaire graf ontwerpen waarin som van de gewichten van aangrenzende faciliteiten worden gemaximaliseerd = feasible blokplan



3. Group Technology

= Celproductie

Even herhalen:

- Combinatie van efficiëntie en flexibiliteit
- Machines die bepaalde productfamilie maken, worden gegroepeerd in een cel
- Cel = verantwoordelijk voor volledige productie van bepaalde familie

A. PPI matrix

= Part-machine Processing Indicator

→ Doel: omvormen PPI matrix tot "blok diagonale" vorm (afgescheiden sets)

Opgelet: NIET alle PPI matrices kunnen zo omgevormd worden → Bottleneck part or bottleneck machine

B. Clusteringalgoritme methodes

i) R&CM algoritme

Step 1: Kies een arbitraire rij die nog niet door een lijn bedekt is. Trek een horizontale lijn doorheen deze rij.

Step 2: zolang er 1'en voorkomen die slechts door 1 lijn worden bedekt, doe

Kies een arbitraire 1 die slechts door 1 lijn wordt bedekt. Indien de lijn horizontaal is, bedek dan de kolom waarin deze 1 voorkomt met een verticale lijn. Indien de lijn verticaal is, bedek dan de rij waarin deze 1 voorkomt met een horizontale lijn.

Step 3: Identificeer de familie en de bijhorende cel

Step 4: Herhaal stap 1 tot stap 3 tot elke rij in de matrix met een lijn bedekt is.

ii) ROC algoritme

(enigste algoritme dat rijen en kolommen fysisch gaat verplaatsen)

Basialgoritme: stel #machines = m, #parts=n

Stap 1: Ken een binair gewicht $BG_j = 2^{m-j}$ toe aan elke kolom j in de PPI matrix.

Stap 2: Bepaal het decimaal equivalent DE_i voor elke rij i: $DE_i = \sum_j 2^{m-j} a_{ij}$

Stap 3: Herschik de rijen zodat de DE_i 's in dalende volgorde staan. Als geen herschikking nodig is: STOP. Anders, ga naar stap 4.

Stap 4: Ken een binair gewicht $BG_i = 2^{n-i}$ toe aan elke rij i in de PPI matrix.

Stap 5: Bepaal het decimaal equivalent DE_j voor elke kolom j: $DE_j = \sum_i 2^{n-i} a_{ij}$

Stap 6: Herschik de kolommen zodat de DE_j 's in dalende volgorde staan. Als geen herschikking nodig is: STOP. Anders: ga naar stap 1.

iii) SC algoritme

(Herkennt wel bottlenecks, kan ermee omgaan)

Berekent **similariteitscoëfficiënt** tussen koppels machines voor groepering machines in cellen.

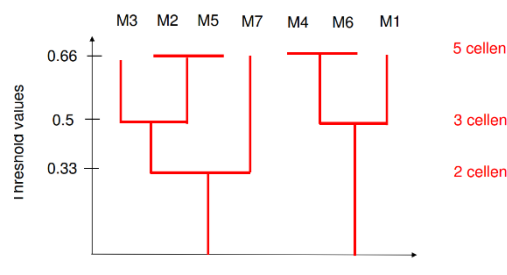
$$S_{ij} = \frac{\sum_{k=1}^n a_{ki} a_{kj}}{\sum_{k=1}^n (a_{ki} + a_{kj} - a_{ki} a_{kj})}$$

Aantal parts dat zowel i en j nodig heeft

Aantal parts dat of i of j nodig heeft, of allebei

Drempelwaardes: $S_{ij} \geq \text{drempelwaarde}$, worden in dezelfde cel geplaatst

→ Dendrogram (vanboven beginnen en zo naar beneden, hier zien we duidelijk 2 cellen/families)



iiii) MP Approach (kunnen opstellen en oplossing interpreteren)

(ook hier kan je de bottlenecks herkennen in de oplossingen)

Groeperen van parts in part families obv dissimilariteitscoëfficiënt

$$d_{ij} = \left[\sum_{k=1}^m |a_{ik} - a_{jk}| \right] \quad \text{Absolute Minkowski metric}$$

→ Doel: Parts zodanig groeperen in families zodat de som van de dissimilarity coefficients geminimaliseerd wordt

■ IP model:

$$\begin{aligned} & \text{Min} \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \\ & \text{st} \\ & \sum_{j=1}^n x_{ij} = 1 \quad \forall i = 1, \dots, n \quad \rightarrow \text{Elk part moet aan precies 1 familie worden toegewezen} \\ & \sum_{j=1}^n x_{jj} = P \quad \rightarrow \text{Aantal gewenste families = P} \\ & x_{ij} \leq x_{jj} \quad \forall i, j = 1, \dots, n \quad \rightarrow \text{Part i kan slechts aan familie j worden toegewezen als deze familie bestaat} \\ & x_{ij} = 0 \text{ or } 1 \quad \forall i, j = 1, \dots, n \quad \rightarrow \text{1 als part i aan familie j toegewezen wordt, anders 0} \end{aligned}$$

4. Assembly Line Balancing

A. Inleiding

Even herhalen:

Flowproductie = assemblagelijnen = product layout
Gebonden lijn versus Ongebonden lijn

We gaan hier verder met de **gebonden lijn**:

- Zonder buffers
- Elk station verantwoordelijk voor uitvoering van bepaald aantal vooraf bepaalde taken
- Productietempo = star: Cyclustijd moet gelijk zijn aan geplande afleveringsinterval
- Tijd is deterministisch
- Hoge automatisatie
- Beperkte taakinhoud
- Zeer repetitief werk

1. Single-model lijn

~~2. Multi-model lijn~~

3. Mixed – model lijn

i) Single-model lijnen

1 product → K taken

T_k = uitvoeringstijd per taak (gekend, deterministisch, onafhankelijk van station)

Totale werkinhoud = som T_k

Er is een technologische volgorde relatie

N eenheden moeten geassembleerd worden in tijdsduur T

Cyclustijd = $c = T/N$

Productietempo = $1/c$

Leegloop op station $i = c - p_i$ (= idle time)

→ minimaliseren van de leegloop = minimaliseren van het aantal stations n

$$\min \sum_{i=1}^n (c - p_i) \rightarrow \min (nc - \sum_{i=1}^n p_i) \rightarrow \min (nc - \sum_{k=1}^K t_k) \\ \rightarrow \min (nc - \text{constante})$$

$n_{\min} = \text{som } t_k / c$ (dan afronden naar hoogste gehele getal)

B. Branch and Bound voor single model lijnen

Even herhalen: Gericht doorzoeken van de oplossingsboom

Diepte = nr van station

Startknooppunt = lege verzameling, diepte = 0

Hoe bepaal je de knooppunten?

1. Som van de duurtijden mag de cyclustijd NIET overschrijden
2. Alle vorige taken moeten eerst toegewezen zijn

Gedomineerde knooppunten

= X ligt dieper of op hetzelfde niveau als y en de taken toegewezen langsheen het pad naar x een deelverzameling zijn van taken toegewezen langsheen het pad naar y.

→ Kunnen enkel leiden tot slechtere oplossingen, je mag ze gewoon weglaten

Benedengrens per knooppunt

ant kiest

$$LB(x) = d(x) + \left\lceil \frac{\sum_{k \in NS} t_k}{c} \right\rceil$$

$d(x)$ = diepte van knooppunt x
 NS = verzameling van nog niet toegewezen taken

De geaccumuleerde leegloop (= hoeveelheid idle time op de tot nu toe bepaalde stations aanwezig is) = $LL(x)$

$LL = nc - \text{som } t_k$

Vertakkingsregels

- Vertakkingsregel:
 - Vertak vanuit knooppunt met kleinste LB
 - Indien tie-break: kies knooppunt met grootste diepte
 - Indien nog tie-break: kleinste geaccumuleerde leegloop

C. Heuristieken voor Single model lijnen

i) Ranked Positional Weight

Stap 1: Geef elke taak x een numerieke score $n(x)$, gelijk aan de som van de duurtijd van x en de duurtijd van alle volgers. Zet stationsnr = $i=1$

Stap 2: zolang nog taken kunnen toegevoegd worden aan station i, doe

Wijz beschikbare taak met hoogste score toe aan station i, indien deze taak binnen de cyclustijd valt en de volgordebepalingenvoldaan zijn.

Werk verzameling beschikbare taken bij: alleen taken waarvan de onmiddellijke voorgangers werden toegewezen zijn beschikbaar.

Stap 3: zet $i=i+1$ en ga terug naar stap 2.

ii) Hoffmann Heuristiek

I) Forward Balance

Genereer voor elk station taaktoewijzing met kleinste leegloop, beginnend vooraan

II) Reverse Balance

Herneem dezelfde procedure maar dan achteraan beginnen

→ oplossing met kleinste stations wordt behouden

D. U-vormige lijnen

= layout die vaak gebruikt wordt voor een cel

Via Branch and Bound

E. Mixed Model Line Balancing

Mixed-model lijn

- Verschillende producttypes worden door elkaar geproduceerd op de lijn zonder omstellingen
- GEEN batches

i) Hoe balanceren?

- Stel gecombineerde precedence diagram op
- Werkinhoud van taakgroepen berekenen op shiftbasis

$$\hat{t}_k = \sum_{j=1}^J N_j t_{jk}$$

N_j = # units te produceren van model j per shift
 t_{jk} = duurtijd taak k op model j

- Taken toewijzen aan stations op shiftbasis

ii) Minimale leegloop minimaliseren

$$\min \sum_{i=1}^n (T - T_i) \Rightarrow nT - \sum_k \hat{t}_k$$

T = duur shift
 T_i = totale work content toegewezen aan station i
 n = aantal stations: te bepalen

I) Voordelen

- Leidt tot taaktoewijzingen die garanderen dat een gegeven taak steeds wordt toegewezen aan een gegeven station
- Als station taken krijgt, is arbeider verantwoordelijk voor taken van alle modellen op dat station

II) Nadelen

- Per model kunnen grote verschillen zijn in work content tussen de stations → taktijd!!!

iii) RPW

= tijd dat na dat stations nog nodig is

- Dalend rangschikken en dan toewijzen aan station

iiii) Repetitive cycli

Hoe lengte van de korts mogelijke cycli bepalen?

- Aantal min per stuk → KGV = cyclustijd

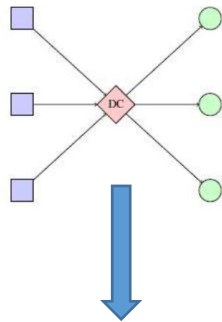
iiii) Schendingen van taktijd wegwerken?

- Taken parallel gaan uitvoeren
- Equipment-paced lines
- Open to the left *versus* open to the right

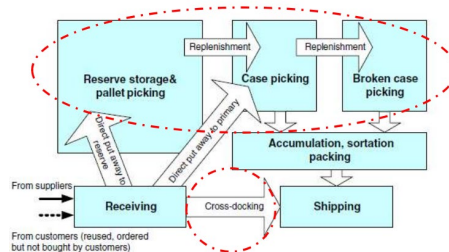
5. Warehousing

A. Warehouse functions and flows

Door een warehouse worden de transportkosten gereduceerd



Alles wordt op 1 plaats verzameld en dan per order naar de klant gebracht.

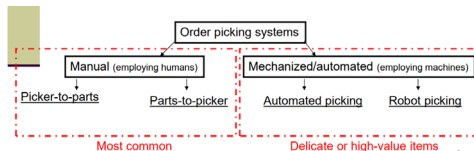


Cross-docking: rechtstreeks van receiving naar shipping; er moet niks tussendoor opgeslagen worden in de warehouse

B. Order picking systems

= Verzamelen van alle individuele items van de stock tot de locaties voor het vervullen van een klantenorder

→ KPI: minimaliseren van de tijd die hiervoor nodig is



→ bij bekijken enkel de **manual**

i) Picker-to-parts

= orderpicker wandelt door de gangen om de parts op te pikken

I) Low level picking

Alles kan men gewoon nemen vanop de grond

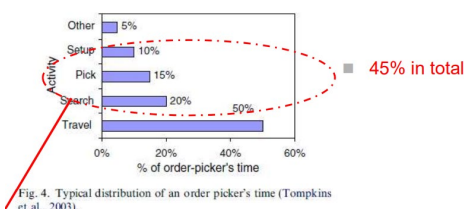
II) High level picking

Verticale hoge stapelrekken
Verplaatsbaar platform

ii) Parts-to-picker

= De picker zelf zit niet in het systeem, automatische systemen brengen de parts tot bij de picker
Vb. KIVA robot

C. Design of order picking systems



These time components of order picking.

Setup + pick + search = 45%

Travel = 55%

How to improve this?

i) How to improve Setup, pick and order time

I) Paper picklists

Alles staat op papier → makkelijk menselijke fouten

II) Pick-by-voice

Pickers krijgen instructies via een headset

III) Pick-by-light

Display on the shelf toont wat en waar ze parts moeten nemen

IV) Pick by HMD

Headmounted display toont aan welke parts en de hoeveelheid

ii) How to improve the travel time?

I) Layout Design

Hoe ga ik mijn magazijn indelen? Waar is het depot? Waar zijn de gangen? ...

II) Storage assignment

1. Random

Elk binnenkomend product wordt random op een vrije plaats gezet

+ Grote plaatsgebruikname

- Werkt alleen in een computer-controlled zone

2. Dedicated

Elk product heeft een vaste locatie

+ Order pickers worden de productlocaties gewoon

- space utilisation is laag, locatie is altijd gereserveerd + maximum inventory

3. Full-turnover storage

Producten worden geplaatst naargelang hun omzet

Meeste omzet → beter bereikbaar, dichterbij depot

COI = volume/ aantal trips

+

- Veel informatie nodig + vraag kan veranderen doorheen de tijd

4. Class-based storage

Producten worden in klassen verdeeld, elke klasse heeft zijn zone in de warehouse

III) Batching

1. Pick-by-Order = discrete picking

Elk order wordt apart verzameld in 1 tour

2. Pick-by-Article = batch picking

Alle artikels voor verschillende orders worden verzameld in 1 tour en verdeeld

Sort while picking or pick and sort

IV) Zoning

Order picking area is verdeeld in zones, elke order picker is verbonden aan een zone.

1. Pick and pass

Een picker begint aan het order, als hij klaar is doet de volgende verder.

2. Parallel picking

Pickers starten samen aan een order en voegen daarna de parts samen.

V) Routing

Geef de items een goede volgorde zodat de pickers een efficiënte route kunnen nemen

Optimal routing methods → computer intensief

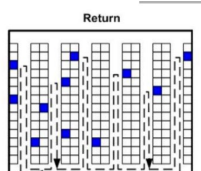
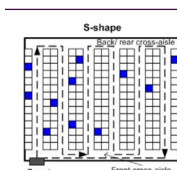
Versus **Heuristic routing methods**

1. S-shape

Men gaat alle gangen door om alle parts op te picken.

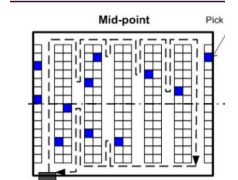
2. Return

Men gaat elke gang in en komt er langs dezelfde kant terug uit.



3. Mid-point

Magazijn wordt in 2 verdeeld, eerst elke gang langs de ene kant ingaan, dan hetzelfde aan de andere kant.



4. Largest Gap

De orderpicker gaat een gang in tot the largest gap.

