

Bestaande ontwerpen van besturingen voor flexibele productiesystemen : een literatuurstudie

Citation for published version (APA):

Janssen, J. H. J. (1988). *Bestaande ontwerpen van besturingen voor flexibele productiesystemen : een literatuurstudie*. (TH Eindhoven. Afd. Werktuigbouwkunde, Vakgroep Produktietechnologie : WPB; Vol. WPA0659). Technische Universiteit Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1988

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Bestaande ontwerpen van besturingen
voor flexibele produktiesystemen, een
literatuurstudie.

W.P.A. rapportnr. 0659

J.H.J. Janssen december 1988

28 maart 1988

Eindstudie-opdracht : J.H.J. Janssen
Afstudeerhoogleraar : Prof.dr.ir. J.E. Rooda
Begeleider : Ir. G.H. Smit
Ir. B.R. Denekamp
Onderwerp : Het ontwerp van een shop controller.

Toelichting

In het kader van het onderzoek naar de besturing van een chipfabriek zullen een aantal shop controllers, die de besturing van shops verzorgen, worden ontworpen.

Het ontwerp van een shop controller dient aan een aantal eisen te voldoen: het ontwerp moet bruikbaar blijven bij een verandering en/of een uitbreiding van de shop. Het moet in de toekomst inpasbaar zijn in een computer integrated manufacturing omgeving en het ontwerp moet in principe ook toepasbaar zijn voor de besturing van andere flexibele manufacturing systemen.

Opdracht

Onderzoek bestaande ontwerpen en de bijbehorende besturingen van flexibele manufacturing systemen. Stel met behulp van deze literatuurstudie een algemeen eisenpakket samen. Leidt uit het ontwerp van een litho shop een specifiek eisenpakket af.

Maak met behulp van dit eisenpakket een ontwerp van de litho shop controller en test dit ontwerp met behulp van de discrete-event simulatie.

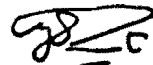
Verslag, etc.:

Het memorandum "Aanwijzigingen voor het afstuderen" is bij de secretaresse verkrijgbaar.

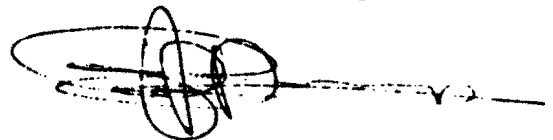
Prof.dr.ir. J.E. Rooda



Ir. G.H. Smit



Ir. B.R. Denekamp



Samenvatting

Flexibele produktie systemen (FPSen) zijn geautomatiseerde produktiesystemen die worden gebruikt voor de vervaardiging van discrete produkten in kleine tot middelgrote series. Ze bestaan in het algemeen uit machines, transportsystemen en een gemeenschappelijke besturing. Binnen de vakgroep WPA wordt momenteel onderzoek verricht naar produktie besturingssystemen voor chipfabrieken. Een chipfabriek kan men opgebouwd denken uit een aantal gekoppelde FPSen. Het doel van dit onderzoek was een algemeen eisenpakket voor een produktie besturingssysteem voor FPSen op te stellen, om hieruit later een specifiek eisenpakket voor een specifiek FPS te kunnen afleiden. Om te weten te komen hoe men dergelijke besturingssystemen heeft gebouwd, is in de literatuur gezocht naar bestaande ontwerpen. Omdat de gedachte bestond dat de problemen zich vooral op het gebied van de scheduling afspelen, is ook gezocht naar scheduling technieken die in de produktieplanning worden gebruikt. Enkele voorbeelden van dergelijke technieken zijn dynamisch programmeren, de 'branch and bound' benadering en integer programmeren. Omdat FPSen momenteel erg in de belangstelling staan wordt hierover veel gepubliceerd. Echter beschrijvingen van bestaande ontwerpen van produktie besturingssystemen voor FPSen komt men slechts sporadisch tegen. En als men artikelen vindt, zijn ze zeer sumier van opzet. Wel wordt er veel geschreven over scheduling en re-scheduling.

Gevolg hiervan is dat men door het lezen van de literatuur niet veel wijzer wordt. Daarom is uitgegaan van een functionele omschrijving van een produktie besturingssysteem, om tot een algemeen eisenpakket te komen. Een produktie besturingssysteem moet aan de volgende eisen voldoen: het moet de functies sturen, besturen en registreren uitvoeren, het moet toepasbaar zijn voor andere FPSen, het moet inpasbaar zijn in een CIM omgeving, het dient modulair van opbouw zijn en moet door de gebruiker makkelijk te bedienen zijn.

Inhoudsopgave

Opdrachtschrijving

Samenvatting 4

1. Inleiding 6
2. Wat zijn flexibele produktiesystemen 7
3. Scheduling technieken 9
4. Produktiebesturing in de praktijk 13
5. Een globaal eisenpakket voor een produktie besturingssysteem 19
6. Conclusies 21

Literatuur 22

Bijlage

- I. Een voorbeeld van dynamisch programmeren 24
- II. Een voorbeeld van de 'branch and bound' benadering 27
- III. Het mathematisch model voor integer programmeren 30

Hoofdstuk 1

Inleiding

In de produktietechniek worden flexibele produktiesystemen (FPSen) steeds meer toegepast. Deze systemen worden toegepast voor de produktie van kleine en middelgrote series. Dat deze systemen erg in de belangstelling staan blijkt onder andere uit de regelmaat waarmee er in de afgelopen jaren is gepubliceerd.

Binnen de vakgroep WPA wordt momenteel onderzoek verricht naar produktie besturingsystemen voor chipfabrieken. Dit zijn fabrieken waar door middel van verschillende fysische processen, elektronische schakelingen op silicium schijven (wafers) worden aangebracht. Het percentage goedgekeurde wafers is sterk afhankelijk van de hoeveelheid stof in de produktieruimtes. Omdat de mens de grootste bron van stof is, is het gewenst de mens uit deze ruimtes te verwijderen.

Een chipfabriek kan men opgebouwd denken uit een aantal gekoppelde FPSen. Het doel van dit onderzoek was om met behulp van een literatuurstudie bestaande ontwerpen van produktie besturingssystemen te onderzoeken, om hieruit een algemeen eisenpakket voor zo'n produktie besturingssysteem te kunnen afleiden.

Hoofdstuk 2

Wat zijn flexibele produktiesystemen

Door Ruissen (1986) wordt een flexibel produktiesysteem als volgt gedefinieerd:

Een flexibel produktie systeem (FPS) is een geautomatiseerd produktie systeem, waarop met een minimum aan handzame interventies verschillende discrete produkten kunnen worden vervaardigd. Het is samengesteld uit een aantal machines, transportmiddelen en een gemeenschappelijke besturing.

Deze definitie is van toepassing op geautomatiseerde produktiesystemen die worden gebruikt voor de fabricage van onderdelen, maar ook op geautomatiseerde produktiesystemen die voor assemblage van produkten worden gebruikt. Dit zijn de zogenaamde flexibele assemblage systemen (FAS). Met behulp van deze produktiesystemen probeert men de volgende doelstellingen te bereiken:

- toename van de produktiviteit.
- vermindering van de produktiekosten.
- vermindering van de voorraadkosten.
- vermindering van het werk in uitvoering.
- verbetering van de kwaliteit.

Het produktiegebeuren kan worden onderverdeeld in een aantal geledingen, zie figuur 1, [Rooda, 1987].

produktie besturing
machine besturing
transducenten
machines
bewerkings processen
materialen

Figuur 1. Geledingen in het produktiegebeuren

Hierin heeft de produktie besturing voornamelijk betrekking op de organisatie rond de verschillende produktie processen, terwijl de andere geledingen meer de technologische aspecten ervan omvatten.

De produktie besturing heeft een drietal functies. Een sturingsfunctie, een besturingsfunctie en een registratiefunctie. De sturingsfunctie van de produktie besturing houdt in: het bepalen van de volgorde waarmee de verschillende produkten aan de verschillende produktie processen zullen worden toegewezen. Hiertoe dient gebruik te worden gemaakt van gegevens die vrijkomen tijdens het functioneren van het FPS. Op welke manier dergelijke volgorde kunnen worden bepaald wordt uitgebreid behandeld in hoofdstuk 3. De besturingsfunctie van de produktie besturing houdt in: de produkten daadwerkelijk op de juiste tijd op de juiste plaats aanwezig te laten zijn. Hiertoe worden aan het transportsysteem opdrachten gegeven. Tot de inhoud van deze functie kan ook worden gerekend de communicatie met de machines, met de transportsystemen en eventueel ook met andere FPSen. De registratiefunctie houdt in: het verzamelen, verwerken en opslaan van gegevens.

Een FPS is samengesteld uit een aantal machines, transportsystemen en een gemeenschappelijke besturing. Machines kunnen zowel CNC-machines zijn als industrieële robots die worden gebruikt voor assemblage-of lasdoeleinden. Zij hebben vaak een eigen gereedschapsmagazijn tot hun beschikking waaruit zij zelfstandig het juiste gereedschap kunnen pakken wat nodig is voor een bepaalde bewerking. Een interface naar een computer netwerk is ook aanwezig. Deze interface is noodzakelijk voor de communicatie met andere machines en computers. Transportsystemen in een FPS kunnen voorkomen in de vorm van AGVs (Automated Guided Vehicles), conveyer systemen en industrieële robots. De gemeenschappelijke besturing bestaat uit computers, terminals, printers, een computernetwerk en bijbehorende software. De software bestaat in principe uit drie gedeeltes. Een besturingssysteem (operating system) met een real-time karakter, een data base management systeem (DBMS) en de applicatie-software.

Hoofdstuk 3

Scheduling technieken

In het voorgaande hoofdstuk is over de sturingsfunctie van de produktie besturing gesproken. In feite houdt die sturingsfunctie niets anders in dan plannen. Een onderdeel van plannen is scheduling. Onder scheduling wordt verstaan:

Het volgen van een strategie die het toewijzen van opdrachten aan een systeem mogelijk maakt zodanig dat men aan een van te voren vastgelegd criterium zo goed mogelijk probeert te voldoen.

Dit impliceert dat de strategie die men volgt niet noodzakelijk tot de beste toewijzing zal leiden. Een systeem heeft in deze context verschillende verschijningsvormen zoals een FPS, een multi-tasking operating system of een start- en land controle-systeem op een vliegveld.

Een manier om scheduling problemen te classificeren maakt gebruik van een notatie met behulp van een viertal parameters, $n/m/A/B$ [French, 1982]. Deze notatie gaat uit van een produktie omgeving. Hierin is:

n is het aantal opdrachten (jobs).

m is het aantal machines.

A beschrijft het soort systeem. Als we te maken hebben met een zogenaamd één-machine probleem ($m = 1$) met onafhankelijke jobs wordt A weggelaten.

F: flow shop.

P: permutatie flow shop. Dat wil zeggen er zijn een aantal jobs die allemaal de zelfde bewerkingsvolgorde hebben en de bewerkingsvolgorde van jobs vóór elke machine is ook het zelfde.

G: de gegeneraliseerde job shop.

B beschrijft het criterium.

De criteria die men gebruikt kunnen weer worden onderverdeeld in:

1. Criteria gebaseerd op de tijdstip van gereedkomen.

F ,

F_{\max} : minimalisering van de gemiddelde doorlooptijd en de grootste door-

looptijd.

\bar{C} ,

C_{\max} : minimalisering van de gemiddelde produktietijd en de totale produktietijd.

Opgemerkt moet worden dat als alle jobs op het zelfde tijdstip beschikbaar zijn voor produktie, dat dan de criteria F_{\max} en C_{\max} het zelfde zijn.

2. Criteria gebaseerd op levertijden.

\bar{L} ,

L_{\max} : minimalisering van de gemiddelde levertijd afwijking en de grootste levertijd afwijking.

\bar{T} ,

T_{\max} : minimalisering van de gemiddelde levertijd overschrijding en de maximale levertijd overschrijding.

n_t : minimalisering van het aantal jobs dat een levertijd overschrijding heeft.

\bar{L} en L_{\max} kunnen negatief zijn. Dit is het geval als de jobs eerder klaar zijn dan de gevraagde levertijd. Bij \bar{T} en T_{\max} is dit niet het geval. Zij zijn altijd positief. Dit onderscheid wordt gemaakt als er misschien premies te verdienen zijn als men eerder kan leveren dan de afgesproken leverdatum. In dat geval zou men gebruik kunnen maken van de criteria \bar{L} en L_{\max} . Zijn er geen premies te verdienen maar krijgt men slecht boetes als men te laat levert dan kan men beter gebruik maken van de criteria \bar{T} en T_{\max} .

3. Criteria gebaseerd op kosten.

\bar{N}_w ,

\bar{N}_u : minimaliseren van het gemiddelde aantal wachtende jobs voor de machines en het gemiddelde aantal nog te voltooien jobs.

\bar{N}_c : minimaliseren van het gemiddelde aantal gereed gekomen jobs.

\bar{I} ,

I_{\max} : minimaliseren van de gemiddelde stilstandtijd en de maximale stilstandtijd van de machines.

Bij de criteria \bar{N}_w en \bar{N}_u minimaliseert men de gemiddelde voorraadkosten van het onderhanden werk. Terwijl men bij het criterium \bar{N}_c minimaliseert naar de voorraadkosten van de gereed gekomen produkten.

Er zullen nu een aantal scheduling technieken worden behandeld in die zin dat alleen de

essentie van de techniek zal worden weergegeven. De eerste twee voorbeelden zijn van het type $n/1/\bar{T}$. Het derde voorbeeld is van het type $n/m/G/\bar{T}$.

Dynamische programmering:

Als eerste zullen er een aantal aannames worden gegeven:

$K = \{J_1, J_2, \dots, J_n\}$, de verzameling van alle jobs.

$J \subset K$, hierin is J de verzameling jobs die al zijn geplaatst.

$J' = K - J$, de verzameling jobs die nog geplaatst moeten worden.

$q = \sum t_j$, waarin t_j de produktietijd voorstelt van job j in J' .

Stel er is een volgorde gecreëerd waarvoor geldt dat alle jobs in J' vooraf gaan aan alle jobs in J . Als de volgorde in J nu optimaal is dan zegt het principe van optimaliteit voor dynamisch programmeren dat het er niet toe doet hoe de jobs in J zijn geplaatst, de jobs in J' moeten ook optimaal worden geplaatst. En dit onder de voorwaarde dat er geen enkele job mag worden geplaatst na q . Een voorbeeld van dit principe wordt nader uitgewerkt in bijlage I. Voor andere voorbeelden en intelligentere algoritmes wordt verwezen naar [Baker, 1974] en [French, 1982].

De 'branch and bound' benadering:

Deze methode maakt gebruik van twee procedures namelijk een 'branch' procedure en een 'bound' procedure. De branch procedure splits het originele probleem in een aantal subproblemen die elkaar wederzijds uitsluiten en een volledige set van problemen zijn die het originele probleem omschrijven, gedeeltelijk opgeloste versies zijn van het originele probleem en minder groot zijn dan het originele probleem. Voor elk subprobleem kan dan weer de zelfde procedure worden gevolgd om de subproblemen verder te splitsen. De bound procedure bepaalt een ondergrens voor het criterium voor elk subprobleem gegenereerd door de branch procedure. De branch en bound methode stelt ons nu in staat om takken die zijn gecreëerd door de branch procedure uit te sluiten voor verder onderzoek door gebruik te maken van de bound procedure. Dit kan gebeuren door te kijken naar de ondergrens van elk subprobleem. Er wordt nu alleen die tak verder onderzocht waarvan de ondergrens het laagste is. Deze techniek wordt jump-tracking genoemd. Een voorbeeld van deze techniek is nader uitgewerkt in bijlage II. Voor andere technieken wordt verwezen naar [Baker, 1974].

Integer programmering:

Integer programmering is een techniek die het scheduling probleem omzet in een aantal lineaire vergelijkingen, ongelijkheden en een te minimaliseren functie. Dit stelsel

van vergelijkingen en ongelijkheden is nu te minimaliseren door gebruik te maken van lineaire programmerings technieken. Heeft men het minimum gevonden dan vertaalt men de oplossing weer terug naar het originele scheduling probleem en men heeft de optimale oplossing gevonden. Op het vertalings probleem wordt verder ingegaan in hoofdstuk 4 en bijlage III.

Van bovenstaande scheduling technieken kan worden gezegd dat ze alle drie veel mathematische bewerkingen vergen. Een classificatie van de moeilijkheid voor scheduling algoritmes is overgenomen uit de combinatoriek, Klee (1980). Men spreekt dan over P en NP scheduling problemen. Wat is een P scheduling probleem.

Een scheduling probleem is van de klasse P als er ooit een algortime is gevonden waar voor geldt dat de functie $p(v)$, een functie voor het aantal mathematische bewerkingen, kan worden beschreven met een polynoom waarvan de graad ten hoogste n is. Dit houdt in dat men een bovengrens kan aangeven voor het maximale aantal mathematische bewerkingen v die men moet uitvoeren om voor een bepaald criterium de optimale oplossing te vinden.

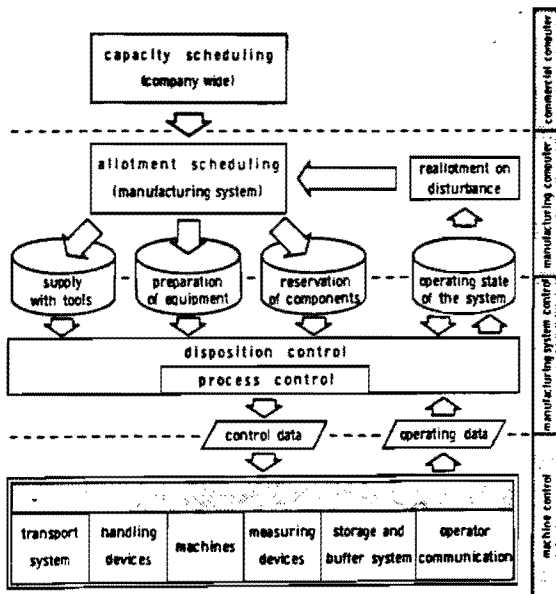
Een scheduling probleem is van de klasse NP als men alleen nog maar algoritmes heeft gevonden waar voor geldt dat de functie $p(v)$ een exponentiële functie is in v .

Men kan van problemen van de klasse NP zeggen dat ze in principe nog niet zijn opgelost. De verwachting is echter dat scheduling problemen die nu van de klasse NP zijn van deze klasse zullen blijven. Voor een meer wiskundige achtergrond wordt verwezen naar [French, 1982]. Voor alle drie hiervoor genoemde methodes geldt dat zij van de klasse NP zijn. Integer programmering is vaak zelfs omslachtiger dan volledige enumeratie.

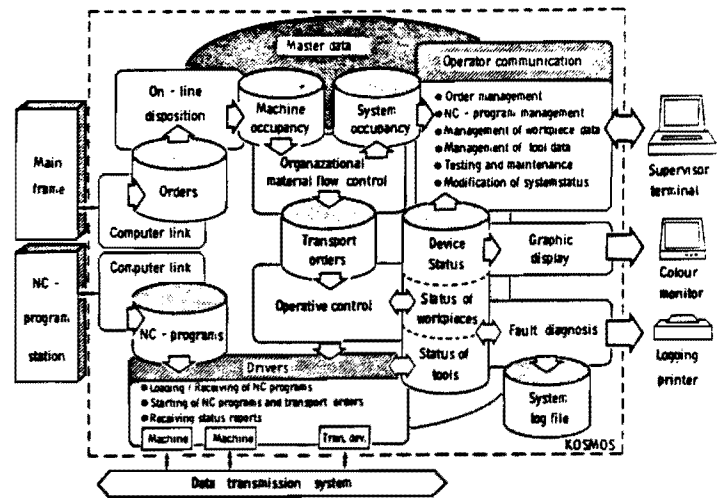
Hoofdstuk 4

Productie besturing in de praktijk

In het hoofdstuk 2 is behandeld wat een FPS is en waaruit het in het algemeen bestaat. In hoofdstuk 3 zijn een aantal scheduling technieken behandeld die worden gebruikt voor zogenaamde statische deterministische problemen. In de praktijk van FPSen is dit zeker niet het geval. Ze zijn dynamisch en stochastisch. In de literatuur worden weinig beschrijvingen aangetroffen van productie besturingen. Wat men wel veel aantreft zijn artikelen over scheduling van FPSen. Ze zijn geschreven vanuit de gedachte dat productie besturing in de praktijk erg moeilijk is en het probleem vaak op de scheduling is terug te voeren. Implementatie van de in de artikelen beschreven scheduling techniek vindt men in de praktijk niet. Men eindigt meestal met enkele simulaties van de beschreven techniek. Als men een artikel vindt over productie besturing in de praktijk wordt hierin alleen de grove structuur weergegeven zoals is aangegeven in figuur 2 en 3, Weck (1986).



Figuur 2. Hiërarchische structuur van een productie besturingssysteem

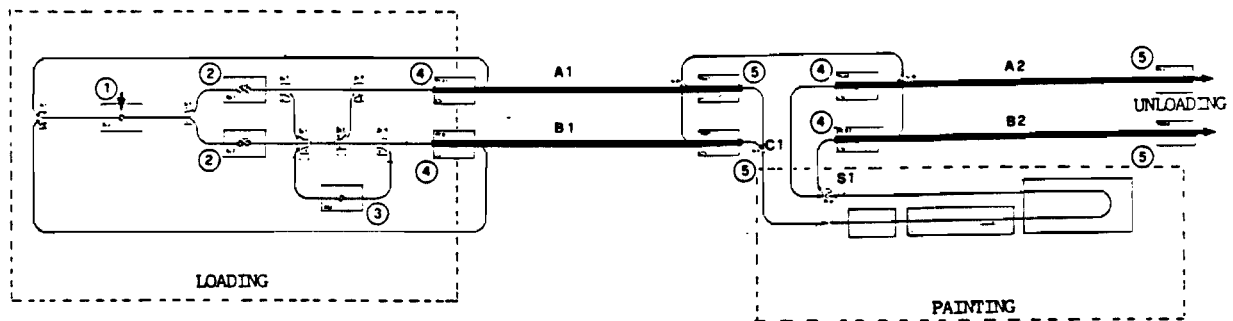


Figuur 3. Software structuur van een productie besturingssysteem

Er zal nu een productie besturingssysteem worden omschreven waarbij wel vrij redelijk is beschreven hoe het in de praktijk werkt.

Het FPS dat wordt behandeld wordt gebruikt voor de assemblage van tractoren bij Fiat

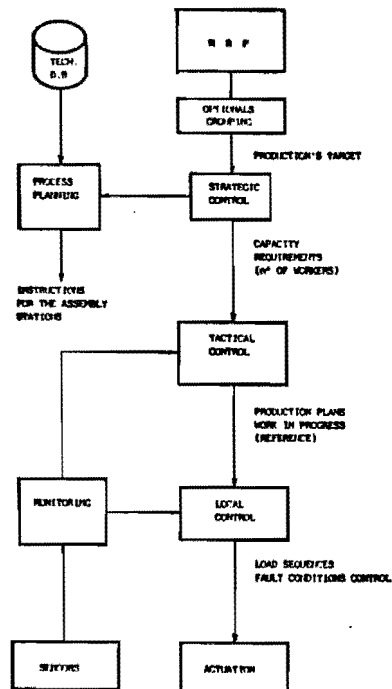
Trattori, de zogenaamde I.E.S.I. assemblage lijn, Testa et al (1985). De lijn is bedoeld om grote aantallen tractoren van verschillende modellen te produceren onder de voorwaarde dat variaties in mix en volume de efficiëntie niet beïnvloeden. De efficiëntie is hier gedefinieerd als de ratio van werkelijke produktietijd en totale produktietijd. De lay-out van het FPS is weergegeven in figuur 4.



Figuur 4. Lay-out van het tractor assemblage systeem

De lijn bestaat uit drie hoofdgroepen, dit zijn laad en samenbouw van de motor en de versnellingsbak, de assemblage lijnen en de lakstraat. De eerste hoofdgroep bestaat uit een viertal werkstations gekoppeld door een automatisch transport systeem dat ook wordt gebruikt om de aandrijfunits te transporteren naar de assemblage lijnen. Op station 1 wordt de versnellingsbak geplaatst. Op station 2 wordt de motor aan de versnellingsbak gekoppeld. Station 3 dient voor andere ondersteunende activiteiten. De stations 4 en 5 dienen voor het beladen en ontladen van de assemblage lijnen. De tweede hoofdgroep bestaat uit een viertal assemblage lijnen. De lijnen a_1 en b_1 vóór de lakstraat en de lijnen a_2 en b_2 na de lakstraat. Elke tractor kan naar elk van de vier assemblage lijnen worden geadresseerd. In de praktijk zijn er dus vier verschillende routings. Het aantal verschillende modellen wat zich in het systeem bevindt wordt beperkt door de aanwezigheid van de verschillende componenten langs de assemblage lijnen. De aandrijfunit, die geplaatst is op een draagblok, beweegt zich met een constante snelheid die wordt bepaald door het aantal arbeiders langs de lijnen. De derde hoofdgroep is de lakstraat. Deze behandelt de tractoren volgens het First In First Out (FIFO) principe. De industriële robots die zich in de lakstraat bevinden krijgen automatisch de programma's behorende bij elk model tractor.

De besturings structuur is weergegeven in figuur 5.



Figuur 5. Besturings structuur van de tractor assemblage lijn

We zien dat de besturings structuur bestaat uit een aantal functionele modules. Dit zijn de strategic control module, de proces planning module, de tactical planning, de local control module en de monitoring module.

Het hoofddoel van de strategic control module is het bepalen van het aantal arbeiders dat nodig is om het hoofdproductieplan te realiseren. Dit wordt gedaan door het balanceren van de routes voor elk model. De tijdhorizon van deze module is een maand. Het hoofdproductieplan is in termen van hoeveelheid verschillende modellen met hun bijbehorende lever-tijd gegeven. De techniek die wordt gebruikt voor het bepalen van het aantal arbeiders is integer programmeren. Voor de vertaling van het probleem naar een lineair programme-rings probleem wordt naar bijlage III verwezen.

De proces planning module bepaalt het aantal basis operaties wat door een arbeider moet worden gedaan. Het probleem wordt opgelost met een algoritme dat de verschillen minima-liseert van het aantal operaties die door de individuele arbeiders moet worden gedaan. Het resultaat is een kaart waarop elke operatie is vermeld die op een specifieke plaats aan de assemblagelijin moet worden gedaan. Deze kaart wordt aan de tractor bevestigd. Hoe het

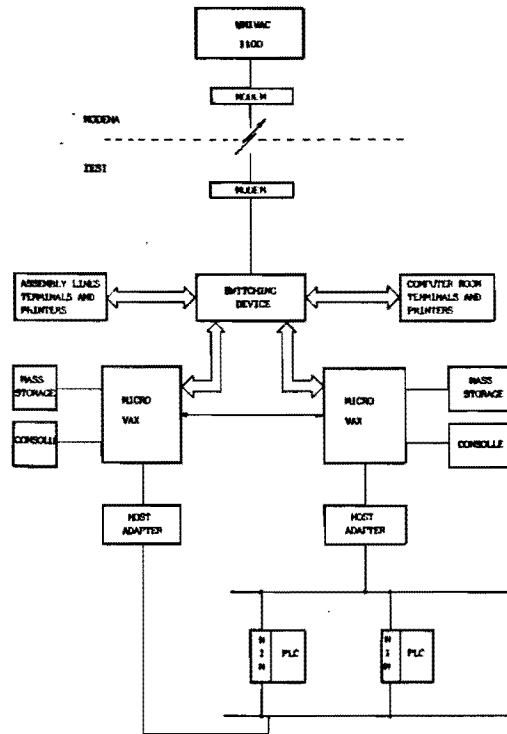
algoritme is opgebouwd wordt niet vermeld.

Het doel van de tactical planning module is het afstemmen van het dagelijks productieplan. Dit als gevolg van uitvallen van lijnen, veranderingen in levertijden, niet aanwezig zijn van onderdelen langs de assemblage lijnen enzovoort. Het eerste gedeelte bepaalt het dagelijks productieplan. De strategie is het zelfde als bij de strategic control module. Het enige verschil is dat het aantal arbeiders al vastligt. Mathematisch wordt dit vertaald in het minimaliseren van de totale produktie tijd. Het tweede gedeelte bestaat uit een analytische simulatie gebaseerd op een queueing-network model. Het definiëert het volume en de mix van het dagelijkse productieplan zodanig dat aan het hoofdproductieplan kan worden voldaan. De hoeveelheid produkten in bewerking wordt zonder een verminderde efficiëntie geminimaliseerd. Hoe de simulator werkt wordt wederom niet vermeld.

De local control module houdt zich bezig met de besturing van het transport systeem en scheduling van de produktie. Een correcte bewerkings volgorde is van het grootste belang om aan het benodigde aantal produkten in bewerking te kunnen voldoen. Een on-line simulatie wordt uitgevoerd om de optimale bewerkingsvolgorde te bepalen. Deze simulator wordt hiervoor met informatie gevoed zoals opgetreden fouten en wachttijden. Deze informatie wordt geleverd door de monitor module. Een andere taak van de local control module is het afhandelen van optredende fouten of afwijkende condities. Dit gebeurt door rescheduling. Local control accepteert ook invoer van operators die de geselecteerde volgorde verwerpen.

De monitor module registreert elke significante gebeurtenis in het systeem en transformeert deze data naar het juiste formaat voor de control modules. Verder geeft het informatie aan de systeem manager over de status van het systeem.

De computer architectuur is weergegeven in figuur 6.



Figuur 6. De computer architectuur van de tractor assemblage lijn

Hierin zien we een SPERRY–UNIVAC 1100 waarop de modules proces planning en strategische control zijn geïmplementeerd. De computer is met behulp van twee modems gekoppeld met twee MICRO–VAX II computers in back–up configuratie. Het voordeel van de back–up configuratie is dat zij beide continu van de status van het systeem worden voorzien. Als nu een computer uitvalt ligt niet meteen het gehele productie proces stil, maar kan er met de andere computer worden doorgewerkt. De communicatie tussen de twee MICRO–VAX II computers en de PLCs wordt door het Local Area Network (LAN) TI–WAY van Texas Instruments verzorgd. Twee PLCs worden gebruikt voor de besturing van het transport systeem en twee voor de besturing van de lakstraat.

De control structuur zoals die hier wordt gebruikt is onderverdeeld in drie hiërarchische niveau's. Dit zijn het strategisch niveau, het tactisch niveau en het lokaal niveau. De onderverdeling is afkomstig van Holstein (1968) en wordt ook gebruikt door van Looveren et al (1986). De tijdhorizon voor elk niveau ligt binnen grenzen vast. Voor het strategisch niveau is dit maanden–jaren. Voor het tactisch niveau is het dagen–weken. En voor het lokaal niveau is dit seconden–uren. Het gevolg van deze indeling is dat men beslissingen neemt voor de toekomst zonder te weten of de werkelijkheid wel zo zal zijn. Dit kan er toe leiden

dat fouten die optreden in het productie proces grote gevolgen hebben op beslissingen die reeds op een hoger niveau zijn genomen. Het hoofd productieplan kan bijvoorbeeld in gevaar komen. Verder worden er een aantal zaken welke voor het functioneren van het FPS essentieel zijn niet beschreven.

Hoofdstuk 5

Een globaal eisenpakket voor een productie besturingssysteem

Een productie besturingssysteem heeft een drietal functies. Een sturing, een besturing en een registratie functie (zie hoofdstuk 2). Bij elk van deze functies kunnen een aantal taken worden gedefinieerd. Bij sturing zijn dit order management, capaciteits planning, materiaalstroom beheersing. Onder order management wordt verstaan het accepteren van orders, order voortgangsbewaking en terugmelding dat orders gereed zijn. Met capaciteitsplanning wordt bedoeld het verwerken van orders zodanig dat rekening wordt gehouden met de beperkte capaciteit van machines, draagblokken en materiaal. Materiaalstroom beheersing is het accepteren, verwerken en bewaken van transportopdrachten. De taken bij de besturings functie zijn fysieke besturing, exceptie afhandeling en gebruikers communicatie. Met fysieke besturing wordt bedoeld het daadwerkelijk opdracht geven aan machines en transportsystemen en het verzenden van data tussen machines onderling en tussen machines en computers. Exceptie afhandeling is het afhandelen van optredende fouten tijdens het functioneren van het FPS. Gebruiker communicatie is het verzorgen informatie uitwisseling met de gebruiker. De taken bij de registratie functie zijn verzamelen, bewerken en opslaan van data.

In de praktijk bestuurt men meestal alleen het transportsysteem, Ruisen (1986). Om te komen tot een volwaardig productie besturingssysteem zou men alle hierboven genoemde taken moeten implementeren. Omdat deze taken aan elkaar zijn gerelateerd is het noodzakelijk om tot een integrale benadering van het probleem te komen. Een methode die in staat is om het probleem integraal te benaderen is de Proces Interactie Benadering [Overwater, 1987]. De methode stelt ons in staat om met behulp van een drietal gereedschappen een FPS te specificeren, te valideren en te implementeren. De gebruikte gereedschappen zijn D86, S84 en ROSKIT (Real-time Operating System KIT). Een ander voordeel van deze benadering is dat men grote delen van de software die men voor de simulatie heeft geschreven, voor de implementatie kan gebruiken. Hierdoor kan men meer tijd aan het specificeren en valideren van het FPS besteden.

Bij het ontwerpen van een produktie besturingssysteem zou men er van uit moeten gaan dat men het ontwerp ook voor andere FPSen zou willen gebruiken. Men zou dus een algemeen model voor een FPS moeten maken. Dit om te voorkomen dat men telkens specifieke oplossingen zoekt voor specifieke problemen. Vanuit dit algemene model zou men de specifieke modellen moeten afleiden. Het voordeel hiervan is dat men gemakkelijker tot integratie van FPSen kan komen. Men heeft dan een kleinere kans tegen het fenomeen eiland-automatisering aan te lopen. Een tweede voordeel is dat men met behulp van deze algemene modellen meer kans van slagen heeft om te komen tot wat men Computer Integrated Manufacturing (CIM) noemt. Verder moet men er van uit gaan dat er tijdens de utilisatie fase van het FPS modificaties aan het FPS zullen plaatsvinden. Om dit zonder problemen te laten verlopen is een modulaire opbouw gewenst. Ook dient men er voor te zorgen dat de gebruiker het FPS makkelijk kan bedienen omdat er waarschijnlijk meer dan één gebruiker van het FPS gebruik zal maken.

Samengevat ziet een eisenpakket voor een produktie besturingssysteem voor een FPS er als volgt uit:

1. Een produktie besturingssysteem moet de taken order management, capaciteits planning, materiaalstroom beheersing, fysieke besturing, exceptie afhandeling, gebruikers communicatie, data verzamelen, data verwerken en data opslaan uitvoeren.
2. Het produktie besturingssysteem moet in principe voor andere FPSen toepasbaar zijn.
3. Het produktie besturingssysteem moet in een CIM omgeving toepasbaar zijn.
4. Het produktie besturingssysteem dient modulair van opbouw te zijn.
5. Het produktie besturingssysteem moet makkelijk te bedienen zijn.

Hoofdstuk 6

Conclusies

Uit het onderzoek naar bestaande ontwerpen met hun besturingen van FPSen is gebleken dat er over dit onderwerp veel wordt gepubliceerd. Dit is niet erg verwonderlijk daar deze systemen erg in de belangstelling staan. Wat men in dergelijke publicaties vindt zijn vooral artikelen over scheduling, rescheduling, simulaties en database systemen. Voorbeelden van dergelijke publicaties in periodieken zijn te vinden in Manufacturing Engineering, Manufacturing Systems en Robotics & Computer – Integrated Manufacturing. Deze artikelen zijn meestal afkomstig van universiteiten of speciale onderzoeks instituten. Dat men hier geen beschrijvingen van functionerende produktie besturings systemen vindt is dus niet zo verwonderlijk. Zij zijn vaak alleen geïnteresseerd in de problemen. Door het lezen van deze artikelen komt men er wel achter wat het probleem is.

Maar ook leveranciers van FPSen geven geen beschrijving over het werkelijke functioneren van hun systeem. Dit uit concurrentie overweging. Een andere reden is dat men zelf ook niet weet hoe men bepaalde problemen op moet lossen. Zoals al in hoofdstuk 5 is vermeld bestuurt men vaak alleen het transportsysteem. Wat het FPS kan wordt in al zijn details uitgelegd. Maar wat het niet kan ondervindt men dan in de praktijk. Produktie besturingssystemen zoals in hoofdstuk 4 vermeld zullen in de praktijk nooit optimaal functioneren omdat men beslissingen die op het sturings niveau zijn genomen op het besturings niveau niet kan terug draaien.

Uit de literatuur is wél gebleken dat de problemen voor produktie besturingssystemen vooral op het gebied van de scheduling zitten. Men probeert dit probleem op te lossen door gebruik te maken van on-line simulaties of door het ontwikkelen van nieuwe scheduling technieken. In hoeverre dit een verbetering zal geven zal de toekomst moeten uitwijzen.

Literatuurlijst

Baker Kenneth R.,

"Introduction to sequencing and scheduling",
John Wiley & Sons, Inc, (1974).

French S.,

"Sequencing and scheduling",
Ellis Horwood Limited, (1982).

Holstein W.,

"Production Planning and Control Integrated",
Harverd Business Review, 46 (3), (1968).

Klee V.,

"Combinatorial Optimization: What is the state of the art?",
Mathematical Operations Research, 5, pag 1–26, (1980).

van Looveren A. J., Gelders L. F., van Wassenhove L. N.,

"A review of FMS planning models",
Modelling and Design of Flexible Manufacturing Systems,
Elsevier, (1986).

Overwater R.,

"Processes and Interactions: an approach to modelling of industrial systems",
Dissertatie T. U. Eindhoven, (1987).

Rooda J. E.,

Inaugurele rede,
T. U. Eindhoven, (1987).

Ruissen E. M.,

"Het besturen van flexibele produktiesystemen",
W.P.A. nr 343, fac. der werktuigbouwkunde,
T. U. Eindhoven, (1986).

Testa P., Vettori A., Corradini F., Naso A.,
"A computer based control system for flexible assembly lines",
Proceedings of the 4th international conference on Flexible Manufacturing Systems",
Elsevier science publishers bv, (1986).

Weck M.,
"Configurable control software for FMS",
Software for discrete manufacturing,
Elsevier science publishers bv, (1986).

Bijlage I

Een voorbeeld van dynamisch programmeren

Het hierna volgende probleem kan als volgt worden omschreven: optimaliseer de volgorde waarin de jobs moeten worden bewerkt zodanig dat de levertijd afwijking minimaal wordt.

De uitgangspunten voor dit probleem, een $4/1/T$ probleem, zijn:

J is de verzameling jobs die zijn geplaatst.

J' is de verzameling jobs die nog geplaatst moeten worden.

$q = \sum t_j$, de produktietijd van de jobs die nog geplaatst moeten worden. Hierbij stelt t_j de produktietijd van job j voor.

$G(J) = \min [g_j(C_j) + G(J - \{j\})]$, de kostenfunctie van de geplaatste jobs.

Hiervoor geldt:

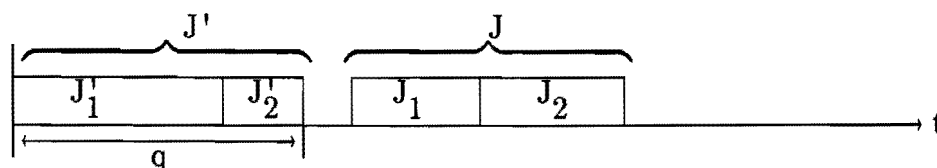
$$g_j(C_j) = w_j(C_j - d_j) = \begin{cases} w_j(q + t_j - d_j) & \text{als } C_j > d_j. \\ 0 & \text{als } C_j \leq d_j. \end{cases}$$

Hierin stelt C_j de doorlooptijd en d_j de leverdatum van job j voor.

Verder geldt: $G(\phi) = 0$ en w_j is de weegfactor voor job j .

Alle jobs zijn op $t = 0$ beschikbaar.

Het een en ander is weergegeven in figuur 7:



Figuur 7. De structuur ten behoeve van dynamisch programmeren

De invoer parameters voor dit voorbeeld zijn:

Job j	t_j	d_j	w_j
1	1	2	1
2	2	7	1
3	3	5	1
4	4	6	1

Uitwerking van het bovenstaande geeft het volgende:

	fase 1			
J	{1}	{2}	{3}	{4}
q	9	8	7	6
$j \in J$	1	2	3	4
g_j	8	3	5	4
$G(J - \{j\})$	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
$G(J)$	8	3	5	4

	fase 2											
J	{1, 2}		{1, 3}		{1, 4}		{2, 3}		{2, 4}		{3, 4}	
q	7		6		5		5		4		3	
$j \in J$	1	2	1	3	1	4	2	3	2	4	3	4
g_j	6	2	5	4	4	3	0	3	0	2	1	1
$G(J - \{j\})$	<u>3</u>	<u>8</u>	<u>5</u>	<u>8</u>	<u>4</u>	<u>8</u>	<u>5</u>	<u>3</u>	<u>4</u>	<u>3</u>	<u>4</u>	<u>5</u>
$G(J)$	9		10		8		5		4		5	

	fase 3											
J	{1, 2, 3}			{1, 2, 4}			{1, 3, 4}			{2, 3, 4}		
q	4			3			2			1		
$j \in J$	1	2	3	1	3	4	1	3	4	2	3	4
g_j	3	0	2	2	0	1	1	0	0	0	0	0
$G(J - \{j\})$	<u>5</u>	<u>10</u>	<u>9</u>	<u>4</u>	<u>8</u>	<u>9</u>	<u>5</u>	<u>8</u>	<u>10</u>	<u>5</u>	<u>4</u>	<u>5</u>
$G(J)$	8			6			6			4		

	fase 4			
J	{1, 2, 3, 4}			
q	0			
$j \in J$	1	2	3	4
g_j	0	0	0	0
$G(J - \{j\})$	<u>4</u>	<u>6</u>	<u>6</u>	<u>8</u>
$G(J)$	4			

Hoe werkt het schema nu. We zien bij fase 1 dat er vier verzamelingen zijn. We nemen het voorbeeld van {3}. Als job 3 als laatste wordt geplaatst geldt $q = t_1 + t_2 + t_4 = 1 + 2 + 4 = 7$. Voor de functie g_j geldt nu $g_3 = w_3 \cdot (q + t_3 - d_3) = 1 \cdot (7 + 3 - 5) = 5$. Voor de functie geldt nu $G(J - \{j\}) = G(\phi) = 0$. Dus volgt $G(J) = g_3 + G(\phi) = 5$. Gaan

we een stap verder en kijken we naar de verzameling $\{1,3\}$ dan zien we dat er hier geldt $q = 6$. Als job 3 nu als laatste is geplaatst betekent dit dat job 1 er voor komt. Voor de functie g_{13} geldt nu $g_{13} = w_1 \cdot (q + t_1 - d_1) = 1 \cdot (6 + 1 - 2) = 5$. Voor $G(J - \{j\})$ geldt nu $G(\{1,3\} - \{1\}) = G(\{3\}) = 5$ en daar uit volgt dat voor $G(J)$ geldt $G(J) = 5 + 5 = 10$. Zo wordt het gehele schema doorgewerkt. Op het einde zien we nu dat als job 1 als eerste wordt geplaatst de maximale levertijdafwijking die op kan treden 4 is, dit als gevolg van een weegfactor 1. Vervolgens kijken we naar de verzameling $\{2,3,4\}$. Job 1 is immers al als eerste geplaatst. We zien nu dat we het beste job 3 als tweede kunnen plaatsen want hiervoor geldt dat $G(J)$ de laagste waarde heeft. Op die manier gaan we steeds verder terug en vinden we de optimale volgorde 1-3-2-4 waar dus voor geldt $\sum T_j = 4$ en dus $\mathbb{T} = 1$.

Bijlage II

Een voorbeeld van de 'branch and bound' benadering

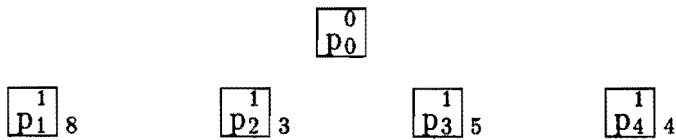
De uitgangspunten voor dit probleem zijn het zelfde als bij bijlage 1. Alleen wordt hier geen gebruik gemaakt van weegfactoren:

Job j	t_j	d_j
1	1	2
2	2	7
3	3	5
4	4	6

Verder wordt gebruik gemaakt van de volgende symboliek:

- $\boxed{p_i^j}$ k i: de index voor de sequence.
 j: de index voor de branch.
 k: de index voor de ondergrens.

Het probleem wordt geïnitieerd met de startwaarde p_0^0 . Vervolgens wordt de eerste tak gecreëerd. Dit houdt in: plaats een job voor de laatst geplaatste job.



Door het creëren van de eerste tak ontstaat er een viertal subproblemen die een gedeeltelijk opgeloste versie zijn van het originele probleem, kleiner zijn dan het originele probleem en elkaar wederzijds uitsluiten. Met behulp van de bound procedure wordt de ondergrens van elk subprobleem bepaald. We krijgen dan bijvoorbeeld voor de levertijd afwijking van het subprobleem p_2^1 het volgende: $\sum t_j = 10$, $d_2 = 7$, $T = 10 - 7 = 3$. Op die manier wordt de ondergrens van elk ander subprobleem bepaald. Vervolgens worden de subproblemen in de zogenaamde actieve lijst geplaatst en wel zodanig dat het subprobleem met de laagste ondergrens bovenaan staat. Het subprobleem wat nu bovenaan in de lijst staat wordt verder onderzocht. Dit is dus het subprobleem p_2^1 .

We krijgen dan het volgende:

$$\begin{array}{ccc} & \boxed{\begin{array}{c} 1 \\ p_2 \end{array}}_3 & \\ \boxed{\begin{array}{c} 2 \\ p_{12} \end{array}}_9 & & \boxed{\begin{array}{c} 2 \\ p_{32} \end{array}}_6 \quad \boxed{\begin{array}{c} 2 \\ p_{42} \end{array}}_8 \end{array}$$

Het subprobleem p_2^1 wordt nu van de lijst verwijderd. De ondergrenzen worden weer uitgerekend en de nieuwe subproblemen worden aan de lijst toegevoegd. Het bepalen van de ondergrens voor bijvoorbeeld p_{32}^2 gaat nu als volgt: $\sum t_j = 8$ (job 2 is immers al geplaatst), $d_3 = 5$. Dit geeft het volgende voor de levertijdoverschrijding: $T = 3 + (8 - 5) = 6$. Het probleem bovenaan de lijst is nu kandidaat om verder te worden onderzocht. Dit is het probleem p_4^1 wat is gecreëerd in de eerste stap.

$$\begin{array}{ccc} & \boxed{\begin{array}{c} 1 \\ p_4 \end{array}}_4 & \\ \boxed{\begin{array}{c} 2 \\ p_{14} \end{array}}_8 & & \boxed{\begin{array}{c} 2 \\ p_{24} \end{array}}_4 \quad \boxed{\begin{array}{c} 2 \\ p_{34} \end{array}}_5 \end{array}$$

De cyclus van verwijderen, ondergrens bepalen en plaatsten in de lijst wordt weer doorlopen met als gevolg dat het subprobleem p_{24}^2 bovenaan de lijst staat. Dit subprobleem is nu kandidaat voor verder onderzoek.

$$\begin{array}{ccc} & \boxed{\begin{array}{c} 2 \\ p_{24} \end{array}}_4 & \\ \boxed{\begin{array}{c} 3 \\ p_{124} \end{array}}_6 & & \boxed{\begin{array}{c} 3 \\ p_{324} \end{array}}_4 \end{array}$$

De cyclus wordt weer doorlopen. Het subprobleem p_{324}^3 komt bovenaan in de lijst te staan. Dus gaan we met het onderzoeken van dit subprobleem verder.

$$\begin{array}{c} \boxed{\begin{array}{c} 3 \\ p_{324} \end{array}}_4 \\ \boxed{\begin{array}{c} 4 \\ p_{1324} \end{array}}_4 \end{array}$$

Het doorlopen van de cyclus heeft nu tot gevolg dat het subprobleem p_{1324}^4 bovenaan in de lijst staat. Dit subprobleem kan niet meer verder worden onderzocht en is dus de gezochte optimale sequence. Deze sequence, 1-3-2-4, heeft een levertijdoverschrijding van 4.

Bijlage III

Het mathematisch model voor integer programmeren

Het doel van de strategic control module is het minimaliseren van het aantal arbeiders langs de assemblagelijnen. Dit wordt gedaan door het balanceren van de routings voor elk model. De techniek die hiervoor wordt toegepast is integer programmeren. In het hierna volgende is alleen de vertaling gegeven naar het mathematisch model. Hoe dit mathematisch model wordt opgelost wordt hier buiten beschouwing gelaten. De vertaling is nu als volgt:

$$\min \sum_{k=1}^4 n_k \quad (1)$$

Het aantal arbeiders gesommeert over elk van de vier assemblagelijnen moet minimaal zijn.

$$\sum_{l=1}^{ndiv} \sum_{j=1}^{nperc(i)} (N_{ij})_l \geq \bar{N}_i \quad \forall i \quad (2)$$

Het te produceren aantal tractoren gesommeerd over alle mogelijke routes en mogelijke subperiodes moet groter of gelijk zijn aan het gevraagde aantal. Dit moet voor elk model i geldig zijn.

$$\sum_{l=1}^d \sum_{j=1}^{nperc(a)} (N_{aj}) \geq (\bar{N}_a)_d \quad \forall a \quad (3)$$

Het te produceren aantal tractoren gesommeerd over alle mogelijke routes en tot de leverdatum moet groter of gelijk zijn aan het gevraagde aantal voor die leverdatum. Dit moet voor elk model a en elke leverdatum d geldig zijn.

$$(N_{ej})_l = 0 \quad \exists e, j \quad l: [f_1, f_2] \quad (4)$$

Gedurende een bepaalde subperiode l is het onmogelijk een bepaald model e volgens route j te fabriceren omdat bepaalde onderdelen niet beschikbaar zijn.

$$(N_{ij})_l = 0 \quad \exists i, j(k) \quad l: [f_1, f_2] \quad (5)$$

Gedurende een bepaalde subperiode l is het onmogelijk een bepaald model i volgens route $j(k)$ te fabriceren omdat er een lijn niet beschikbaar is. De route j is een functie van lijn k .

$$\sum_{i=1}^{nmod} \sum_{j=1}^{nperc(i)} (N_{ij})_l \cdot t_{ijk} \leq \bar{n}_k \cdot T_l \quad (6)$$

$$k = 1, 2 \quad l: [0, \tau]$$

De gevraagde produktie capaciteit voor alle modellen i , volgens alle mogelijke routes j gedurende subperiode l moet kleiner of gelijk zijn aan het produkt van het aantal arbeiders van de vorige maand en de gemiddelde doorlooptijd. Dit geldt alleen voor de lijnen 1 en 2. Dit wordt gedaan om een vloeiende overgang te verkrijgen van de ene ploeg naar de andere ploeg.

$$\sum_{i=1}^{n_{\text{mod}}} \sum_{j=1}^{n_{\text{perc}(i)}} (N_{ij})_l \cdot t_{ijk} \geq n_k \cdot T_l \quad (7)$$

$$k = 3,4 \quad \forall l$$

$$k = 1,2 \quad l: [\tau, T]$$

De gevraagde produktie capaciteit voor alle modellen i , volgens route j gedurende subperiode l moet kleiner of gelijk zijn aan het produkt van het aantal arbeiders en de tijdsduur van de subperiode l . Voor de lijnen 3 en vier geldt dit voor alle subperiodes l . Voor de lijnen 1 en 2 geldt dit voor de periode $[\tau, T]$ omdat de periode $[0, \tau]$ al in (6) in rekening is gebracht.

Lijst van gebruikte symbolen.

- n_k = aantal arbeiders.
- \bar{n}_k = aantal arbeiders van de laatste maand.
- \bar{N}_i = maandelijks te produceren hoeveelheid tractoren van model i .
- $(\bar{N}_a)_d$ = hoeveelheid te produceren tractoren met leverdatum T_d .
- $(N_{ij})_l$ = werkelijke produktie van model i volgens route j in subperiode l .
- n_{mod} = aantal verschillende modellen.
- n_{div} = aantal verschillende subperiodes per maand.
- $n_{\text{perc}(i)}$ = aantal verschillende routes voor model i .
- t_{ijk} = effectief benodigde produktietijd voor model i volgens route j op lijn k .
- T_l = tijdsduur van subperiode l .
- τ = gemiddelde doorlooptijd van lijn 1 en 2.
- f_1 = begin van de periode waarop een lijn en/of onderdelen niet beschikbaar zijn.
- f_2 = einde van de periode waarop een lijn en/of onderdelen niet beschikbaar zijn.
- d = index voor de leverdatum.