

*Een onderzoek betreffende een dynamisch programmeringsalgorithme voor het N-produkten één machine probleem.**

*P. van Schuylenburg ***

Samenvatting:

In dit artikel wordt een nieuwe oplossingsmethode voor een categorie van N-produkten één machine problemen met deterministische vraagstructuur gepresenteerd.

Het principe van deze methode is om eerst voor ieder produkt afzonderlijk een optimaal produktieplanningsschema te berekenen en m.b.v. deze informatie voor alle produkten samen een zo goed mogelijk schema te vinden. Deze aanpak heeft het voordeel dat de rekentijd, nodig om het probleem met een computer op te lossen, niet meer exponentieel, zoals bij een direkte simultane aanpak het geval zou zijn, maar ongeveer lineair is in het aantal produkten (N). Hierdoor is het mogelijk om ook voor problemen met een groot aantal produkten (≥ 15) en een groot aantal perioden (≥ 40) oplossingen te vinden.

Naast de bespreking van deze oplossingsmethode zullen ook nog enige testresultaten gegeven worden om enig idee van de kwaliteit (hier gedefinieerd als afwijking van het echte optimum) te verkrijgen.

Trefwoorden: Produktieplanning, deterministische vraag, dynamische programmering.

* Met dank aan P. van Beek en H.C. Hoefnagels voor hun belangrijke bijdrage aan dit onderzoek.

** N.V. Philips Gloeilampenfabrieken, Centre for Quantitative Methods, gebouw VN-708, Eindhoven

Voor een fabriek is een produktieplanningssysteem ontworpen voor het assembleren van apparaten. De probleemstelling was als volgt: Met behulp van één assemblageband moesten N typen apparaten in elkaar gezet worden. Om praktische redenen werd gewerkt met halve dagen als produktieperioden.

Gedurende een halve dag mocht de assemblageband slechts óf stilstaan óf voor 100% in bedrijf zijn, terwijl maar één type tijdens zo'n periode geassembleerd kon worden. Verder moest de voorraad van elk type (al geassembleerd) apparaat groter of gelijk aan nul zijn.

De vraag naar de typen werd deterministisch verondersteld omdat men beschikte over betrouwbare schattingen.

Andere redenen om een deterministische vraag te veronderstellen zouden bijvoorbeeld het werken met orders of langdurige contracten kunnen zijn.

Het aanpassen van de assemblageband om een type in elkaar te zetten, vergde een verwaarloosbare hoeveelheid tijd, maar bracht wel kosten met zich mee. Wanneer type j geassembleerd moest worden, terwijl in de vorige periode een ander type behandeld werd, bracht het omschakelen kosten F^j met zich mee.

De hoeveelheden van type j per assemblageperiode (hier een halve dag) bedroegen een vaste waarde, aangegeven met K^j , vanwege de veronderstellingen van het voor 100% in bedrijf moeten zijn van de band in een assemblageperiode en het gegeven dat maar één type per periode behandeld kan worden. De voorraadkosten werden recht evenredig verondersteld met de gemiddelde voorraadhoogte.

Het streven was een assemblageschema te vinden dat de totale voorraad-, omschakel- en assemblagekosten * minimaliseert, terwijl aan de vraag voldaan kan worden zonder met backorders te werken.

* De assemblagekosten worden hier onafhankelijk van de grote van de serie verondersteld en in de tijd niet veranderlijk, zodat deze kosten verder niet meegerekend hoeven te worden.

Het probleem laat zich nu als volgt formuleren:

$$\begin{aligned}
 &\text{Minimaliseer} && \left\{ \sum_{t=1}^T \sum_{j=1}^N \left[F^j \times \delta(P_t^j, P_{t-1}^j) + \frac{\alpha^j}{2} \times (I_t^j + I_{t-1}^j) \right] \right\} \\
 &P_1^1 \dots P_1^N \dots P_T^1 \dots P_T^N \\
 &\text{onder} && \sum_{j=1}^N P_t^j \leq 1 && \text{voor } t=1, 2, \dots, T \\
 &&& P_t^j = 0, 1 && \text{voor } t=1, 2, \dots, T \\
 &&& && \text{en } j=1, 2, \dots, N \quad (1) \\
 &I_t^j = I_{t-1}^j - D_t^j + P_t^j \times K^j \geq 0 && \text{voor } t=1, 2, \dots, T \\
 &&& \text{en } j=1, 2, \dots, N
 \end{aligned}$$

waarbij

- F^j de set-up kosten voorstelt
- δ een functie is welke de waarde 1 aanneemt wanneer de machine omgeschakeld moet worden en 0 wanneer dit niet het geval is
- α^j de voorraadkosten voorstelt per eenheidprodukt per periode van produkt j
- I_t^j de voorraadhoogte van produkt j aan het einde van periode t is
- P_t^j aangeeft of produkt j in periode t niet of wel geproduceerd wordt via respectievelijk de waarden 0 en 1
- T het aantal perioden tot aan de horizon voorstelt
- N het aantal produkten weergeeft.
- K^j inhoudt de grootte van de produktieserie in één periode van produkt j (een gegeven waarde)
- D_t^j de vraag naar produkt j in periode t voorstelt
- I_0^j de beginvoorraad voor periode 1 is (gegeven)
- I_T^j de eindvoorraad van periode T is (ook gegeven)

In deze probleemstelling kan een N -produkten één machine probleem herkend worden. Daarom is in het bovenstaande gekozen voor de term produceren i.p.v. assembleren.

Het in (1) geformuleerde probleem blijkt echter voor grotere waarden van T en N niet direkt oplosbaar. Dit wordt veroorzaakt door het exponentieel zijn van het aantal combinaties van voorraad-hoogten der verschillende produkten in het aantal produkten (N).

In de door ons geraadpleegde literatuur over N-produkten één machine problemen konden wij voor dit specifieke geval geen adequate oplossingsmethoden vinden, welke een situatie met een groter aantal produkten (≥ 15) en een behoorlijk aantal perioden (≥ 40) aan zouden kunnen.

Dit was de reden om zelf voor dit probleem een oplossingsmethode te ontwikkelen. Deze methode is de twee-fasen-methode genoemd, omdat er twee fasen zijn waarin produktieschema's worden berekend. Deze twee fasen zijn met elkaar verbonden d.m.v. een effeningsprocedure. Een bespreking van de werking van deze oplossingsmethode volgt hieronder.

In de eerste fase wordt voor ieder produkt afzonderlijk het optimale produktieschema bepaald. Dit komt neer op het overgaan van een N-produkten één machine probleem naar N-één produkt één machine problemen. Oplossing van deze N problemen geeft N optimale produktieschema's voor deze situatie.

Voor produkt j kunnen we dit als volgt formuleren:

Fase I

$$\text{Minimaliseer } \left\{ \sum_{t=1}^T \left[F^j \times \delta(P_t^j, P_{t-1}^j) + \frac{\alpha^j}{2} \times (I_t^j + I_{t-1}^j) \right] \right\}$$

$P_1^j, \dots, P_T^j$

(2)

$$\text{onder } P_t^j = 0, 1 \quad t=1, 2, \dots, T$$

$$I_t^j = I_{t-1}^j + P_t^j \times K^j - D_t^j \geq 0 \quad t=1, 2, \dots, T$$

Dit probleem wordt opgelost via de dynamische programmeringstechniek.

De objectfunctie blijkt namelijk separeerbaar te zijn in de tijd, (d.w.z. in de verschillende perioden) op de variabelen I_{t-1}^j en P_{t-1}^j na. Deze variabelen worden derhalve als toestandsvariabelen gekozen. (Voor een beschrijving van de dynamische programmeringstechniek zie [2]).

De P_{t-1}^j kan zonder complicaties van rekentechnische aard eenvoudig als toestandsvariabele worden opgenomen daar deze variabele slechts de waarde 0 of 1 kan aannemen.

Met I_{t-1}^j ligt dit echter minder eenvoudig. De voorraad blijkt niet met vaste stapjes te veranderen. Voor een mogelijk voorraadverloop zie figuur 2 (gestippelde lijn).

Wanneer we ons echter meer in dit probleem verdiepen zien we dat er per periode maar een beperkt aantal mogelijke niveaus zijn voor de voorraad, terwijl het verschil tussen die voorraadviveaus een geheel aantal maal K^j bedraagt. Figuur 1 geeft voor een aantal perioden de voorraadmogelijkheden weer.

We hebben dit het variabele rooster genoemd. Variabel omdat de afstand van het onderste punt tot de nul-as niet hetzelfde is voor al de perioden en een rooster omdat de afstanden tussen twee opeenvolgende punten per periode constant is (nl. K^j) en hetzelfde voor al de perioden. Dit is het gevolg van de hoogstens twee mogelijkheden die we in een periode hebben nl. niet produceren of een hoeveelheid K^j produceren.

In de dynamische programmeringsformulering ziet het probleem (2) er als volgt uit:

$$V_T^j(P_T^j, I_T^j) = \begin{cases} 0 & \text{als } I_T^j \text{ de hoogte van het gewenste eindvoorraad} \\ & \text{roosterpunt is} \\ \infty & \text{voor de andere gevallen} \end{cases}$$

$$V_{t-1}^j(P_{t-1}^j, I_{t-1}^j) = \min_{P_t^j} \left[F^j \times \delta(P_t^j, P_{t-1}^j) + \frac{\alpha^j}{2} \times (2I_{t-1}^j + P_t^j \times K^j - D_t^j) + V_t^j(P_t^j, I_{t-1}^j + P_t^j \times K^j - D_t^j) \right]$$

onder $P_t^j = 0, 1$

Door V_t^j de waarde oneindig te geven wanneer de eindvoorraad in periode t negatief is, wordt impliciet verzekerd dat de voorraad positief blijft, tenminste als er inderdaad een toegelaten oplossing bestaat. Uitgaande van de horizon T kunnen we nu voor alle relevante waarden voor de produktie en de voorraadroosterpunten de waardefuncties van de T -de t/m de eerste periode bepalen. Wanneer we het beginvoorraadniveau weten kunnen we met behulp van de waardefuncties V_t^j een optimaal schema voor deze situatie vinden.

Nu blijken deze schema's doorgaans niet simultaan verwezenlijkt te kunnen worden. Derhalve is een effeningsprocedure ontworpen om toch een toegelaten tijdschema voor de machine te verkrijgen. Het onderstaande voorbeeld voor een twee-produkten situatie moge het één en ander verduidelijken.

perioden	1	2	3	4	5	6	7	8	9	10	11	12
produkt 1	0	1	1	0	0	0	1	1	0	0	0	1
produkt 2	0	0	1	1	1	0	0	1	1	0	1	0
geaggregeerd	0	1	2	1	1	0	1	2	1	0	1	1

In het bovenstaande tableau stelt een 0 voor dat er geen produktie is van het betreffende produkt in die periode en een 1 dat er wel produktie is.

In periode drie en acht in het bovenstaande voorbeeld treden moeilijkheden op omdat er geen twee produkten tegelijk in één periode op de machine geproduceerd kunnen worden.

Derhalve moet dit schema geëffend worden. In periode acht zijn er twee mogelijkheden nl. verschuiven van één eenheid produktie naar periode zes of tien. Welke van deze twee mogelijkheden het beste is hangt af van de verhouding voorraadkosten en omschakelkosten en van informatie betreffende de vraag. Als criterium voor het effeningsproces is het minimaliseren van een bepaalde objectfunctie gebruikt. Een tweetal van deze functies zal in het onderstaande nog ter sprake komen.

Fase II

De tweede fase gaat uit van het effeningsschema met nullen en enen en wijst de produktie van de produktieperioden toe aan de produkten. Dit gaat d.m.v. het per periode minimaliseren van de extra kosten welke ontstaan door het afwijken van de fase I schema's.

De totale fase I kosten kunnen nl. gezien worden als een (niet te bereiken) ondergrens voor de fase II kosten.

Het toewijzen van de produktie aan de produkten gaat op de volgende wijze (voor periode t):

$$\text{Minimaliseer } \left\{ \sum_{j=1}^N \left[F^j x \delta(P_t^j, P_{t-1}^j) + \frac{\alpha^j}{2} x (2I_{t-1}^j + P_t^j x K^j - D_t^j) + V_t^j (P_t^j, I_{t-1}^j - D_t^j + P_t^j x K^j) \right] \right\}$$

$P_t^1 \dots P_t^N$

onder $\sum_{j=1}^N P_t^j = 1$ (de perioden waar $\sum_{j=1}^N P_t^j = 0$ hoeven niet op deze manier behandeld te worden)

$$P_t^j = 0, 1 \text{ voor } j=1, 2, \dots, N$$

Dit moet gedaan worden voor perioden $1 \leq t \leq T$.

Het essentiële idee van deze heuristiek is het gebruik van de V_t^j waardefuncties uit de eerste fase in de tweede fase.

De gedachte hierachter is dat we zo goed mogelijk de fase I schema's moeten volgen, zodat we dicht bij de (meestal niet haalbare) fase I kosten uitkomen. Zie bijvoorbeeld ook, van Beek [1] en Wijngaard [3] waar een dergelijk idee gebruikt wordt.

Nu kunnen we ook zien waarom in de eerste fase gekozen is voor het van achter (de horizon) naar voren (de eerste periode) ontwikkelen van de waardefuncties V_t^j .

Wanneer voor een bepaald produkt van het optimale fase I schema afge-
weken moet worden, kunnen we voor de dan ontstane nieuwe situatie -
met bijbehorende waarden voor de produktie en voorraadhoogte - weer
het daarvoor beste schema berekenen m.b.v. de berekende waardefuncties V_t^j .
(Zie ook ononderbroken lijn in figuur 2.)

Ook is de werking van deze heuristiek onderzocht.

Omdat wij de overtuiging hadden dat het twee-produkten-geval een 'worst case' zou zijn, is aan deze situatie veel aandacht besteed. Deze overtuiging kwam voort uit de redenering dat wanneer in de tweede fase van de schema's uit de eerste fase moet worden afgeweken bij meer produkten-situaties het aantal mogelijkheden om te kiezen groter is, en derhalve de afwijking van het fase I schema doorgaans qua kosten kleiner zal zijn. Dit geldt in het bijzonder wanneer door de effeningsprocedure een minder goed effeningsschema wordt gekozen. Dit zal bij een meer produkten-situatie over het algemeen minder invloed hebben dan bij een twee-produkten-situatie.

In het begin van dit artikel is geschreven dat probleem (1) voor grote T en N niet simultaan oplosbaar is. Voor het twee-produkten-geval is dit echter nog wel binnen de rekentechnische mogelijkheden.

Oplossen van twee-produkten-gevallen via de simultane aanpak leidt dan tot het minimum voor deze situaties.

Hiervan is dankbaar gebruik gemaakt, omdat de oplossingen verkregen m.b.v. de twee-fasen-algoritme nu vergeleken kunnen worden met het minimum voor het twee-produkten-geval, wat juist de 'worst case' is.

Hiervoor zijn een aantal problemen geconstrueerd, waarbij van de volgende redenering uitgegaan is:

Construeer zo moeilijk mogelijke problemen door bijvoorbeeld lastige vraagschema's te nemen en te werken met extreme verhoudingen voor voorraad en omschakelkosten. Wanneer voor deze moeilijke situaties nog redelijke antwoorden gevonden worden, zal de oplossingsmethode voor 'normale' situaties zeker goed werken.

De resultaten met twee effeningsprocedures zullen gegeven worden

$$\text{Effening A : Min.}_{P_1 \dots P_T} \left\{ \sum_{t=2}^T \left[b \cdot (P_t - P_{t-1})^2 + c \cdot (P_t - \hat{P}_t)^2 \right] \right\}$$

$$\text{onder } P_t = 0,1 \quad t=1, \dots, T \quad b, c \geq 0$$

Met P_t is de geaggregeerde produktie in periode t en $\hat{P}_t$ is de geaggregeerde produktie in periode t volgens fase I schema's.

$$(\hat{P}_t = \sum_{j=1}^N P_t^j \text{ met } P_t^j \text{ de optimale oplossing in fase I}).$$

Term I is opgenomen om in geval van hoge omschakelkosten grote series te kunnen produceren en de tweede term zorgt voor het zo goed mogelijk volgen van de fase I schema's.

Met de geconstrueerde problemen vonden we dat:

15% van de beschouwde problemen een afwijking van het optimum had van kleiner dan 1% en 80% een afwijking van kleiner dan 7%:

$$\text{Effening B : Min. } \left\{ \begin{array}{l} T \\ \sum_{t=2} \left[b \cdot (P_t - \hat{P}_t)^2 + c \cdot (\bar{I}_t) \right] \end{array} \right\}$$

$P_1 \dots P_T$

$$\text{onder } P_t = 0,1 \quad t=1, 2, \dots, T \quad b, c \geq 0$$

Waarbij $\bar{I}_t$ het aantal keren produktie voorstelt tot en met periode t , wat meer toegewezen is dan strikt noodzakelijk zou zijn om in perioden 1 t/m t aan de vraag naar ieder produkt te kunnen voldoen.

Hierin zit derhalve informatie omtrent de vraagzijde verwerkt.

53% van de problemen opgelost via de twee-fasen-methode gaf een afwijking t.o.v. het optimum te zien welke kleiner was dan 1% terwijl 80% van de problemen een afwijking gaf kleiner dan 5%.

Zeer redelijke resultaten en daarom menen wij ook hoge verwachtingen te mogen koesteren omtrent de resultaten wanneer deze methode ook meer op bestaande situaties zal worden toegepast.

Hiernaast zijn om de werking van de tweede fase apart te onderzoeken ook nog de optimale produktieschema's, resulterende uit het oplossen via de simultane aanpak ($N=2$), vertaald in een 0-1 schema en direkt in de tweede fase ingevoerd in plaats van de effening.

Bij alle geconstrueerde problemen zijn toen geen afwijkingen gevonden tussen het optimum voor deze problemen en de via de tweede fase berekende oplossing.

Hieraan kunnen we de volgende conclusies verbinden:

1. Dit kan gezien worden als een rechtvaardiging voor het gebruik van de waardefuncties, berekend in de eerste fase, in de tweede fase.
(Dit is juist het essentiële idee achter deze heuristiek.)
2. De oorzaak van de afwijkingen tussen de oplossingen verkregen via de twee-fasen-methode en de simultane aanpak is hoofdzakelijk het niet kunnen vinden door de effeningsprocedure van de 'beste' effening. Daarom is er met verschillende objectfuncties voor de effening gewerkt waarvan effening B tot nu toe de beste resultaten gaf.

Hiernaast is ook met een aanpak geëxperimenteerd, waarbij de effeningsprocedure en de tweede fase samen werden genomen. Dit heeft als voordeel dat tijdens het effeningsproces met het effect van de effening op de tweede fase rekening gehouden kan worden.

Ook hebben we een techniek ontworpen waarmee problemen met omstelkosten via de twee-fasen-methode kunnen worden opgelost. De omstelkosten O_{ij} worden dan via bepaalde formules omgerekend naar de F_i 's, waarbij het in de vorige periode geproduceerde produkt i niet van belang is voor de hoogte van de afstelkosten voor de machine voor produkt j .

Verder zijn er nog ideeën om via deze oplossingsmethode goede begin- en eindvoorraden te berekenen, wat erg belangrijk is in het geval van een op de continuïteit gericht productieproces.

Referenties:

1. P. van Beek;

An Application of Dynamic Programming and the
HMMS Rule On Two-Level Production Control.

Zeitschrift Für Operations Research,
Band 21, 1977, Seite B 133 - B 141

2. H.M. Wagner;

Principles of Operations Research

Prentice / Hall, Englewood Cliffs 1972

3. J. Wijngaard;

Decomposition for Dynamic Programming in Production and
Inventory Control.

Engineering and Process Economics 4 (1979), pp 385-388