

**EEN HEURISTISCH PROGRAMMA  
VOOR DE OPTIMALISATIE VAN DE LADINGSAMENSTELLING  
IN DE GLOEIAFDELING VAN EEN STAALFABRIEK (\*)**

door ir. D. VERCRUYSEN  
*SIDMAR, Gent.*

**INHOUD**

1. Inleiding.
2. Situering en werking van de gloeiafdeling in een geïntegreerd staalbedrijf.
3. Berekeningsmethode van de gloeitijd.
4. Bespreking van de doelfunctie.
5. Maximalisatie van het gemiddeld gewicht.
6. Vooruitrekenen der rolgloeitijden.  
Maximalisatie van de ovenproductie per uur.
7. Objektieve vergelijkingspunten.
8. Heuristische procedure.
  81. Volgorde der bewerkingen.
  82. Algemene voorwaarde waaraan een heuristische procedure moet voldoen.
  83. Het manueel uitvoeren van procedures.
9. Heuristische programmatie.
  91. Het heuristisch programma.
  92. De Q-methode, de verwerpsprocedure.
  93. De kombinatorische methode.
10. Besluit.  
Annexe A : Symbolen.  
Annexe B : Bibliografie.

**1. Inleiding.**

Meer en meer doet het moderne bedrijfsbeheer beroep op wiskundige beslissingstechnieken zoals de wiskundige programmatie, de wachttijnen-theorie, de speltheorie en de netwerkplanning.

---

(\*) Op 29 januari 1968 werd door de auteur van dit artikel een voordracht gehouden over hetzelfde onderwerp in het kader van SOGESCI.

Voor de opkomst van de computer leverde het gebruik van deze technieken rekentechnisch gezien enorme moeilijkheden op zodat ze dan ook weinig toegepast werden. Veel bedrijven beschikken nu echter over een computer die hun toelaat met behulp van bibliotheekprogramma's, geleverd door de computerfabrikant, deze beslissingstechnieken te gebruiken en te verwerken.

Heel wat problemen lenen zich echter niet tot bovenstaande technieken zodat we in die gevallen aangewezen zijn op heuristiek.

Met heuristiek, afgeleid van het grieks *εὕρισκω* (vinden, zoeken) wordt bedoeld, principes of methodes die bijdragen tot de vereenvoudiging van het zoeken naar een oplossing van een complex probleem.

Heuristiek is een veel gebruikt middel, onder de vorm van vuistregels, bij het dagelijks oplossen van problemen van complexe aard, bijvoorbeeld het bepalen van de routing voor vrachtwagens van een distributiefirma.

In problemen waar geen algoritme bestaat is men aangewezen op heuristiek, in afwachting dat er een algoritme gevonden wordt, daar waar echter wel een algoritme voorhanden is kan heuristiek ook interessant zijn rekentechnisch gezien. Heuristiek is vooral van toepassing bij kombinatorische problemen zoals het travelling salesman probleem (Bibl. 1), het trim-probleem, het gloeiovenprobleem en andere.

Bij deze problemen is het aantal oplossingsmogelijkheden voor een basispopulatie van  $n$ -elementen, meestal van de grootte-orde van  $(n - a)$ -fakulteit.

Bijvoorbeeld in het travelling salesman probleem, waar gevraagd wordt de kortste weg te vinden die  $n$ -steden onderling verbindt, is het aantal mogelijke wegen  $(n - 1) !$

Het exhaustief oplossen van dergelijk probleem vereist het genereren van de verschillende oplossingen, per oplossing het berekenen van een meet-criterium en het kiezen van de beste oplossing. Het is evident dat van zodra de basispopulatie groot is het rekentechnisch onmogelijk wordt de exhaustieve methode te gebruiken.

Om de gedachten te vestigen  $100 ! = 3 \cdot 10^{157}$ , veronderstellen we dat we beschikken over een computer die  $10^6$  oplossingen kan genereren en evalueren per seconde, dan zou de eindoplossing  $3 \cdot 10^{114}$  jaren vereisen.

Bij kombinatorische problemen zal een heuristische procedure, dit is een procedure die de heuristische regels definieert, een groot deel van de  $n !$  mogelijkheden elimineren en enkel een beperkt aantal combinaties onderzoeken. Weliswaar is de optimale oplossing niet verzekerd maar benadert de beste oplossing de optimale.

Wat nu ons probleem betreft komt dat neer op het volgende. Dagelijks staat de verantwoordelijke van een gloeiafdeling voor een belangrijk combinatorisch probleem, hij moet uit een stock van 50 à 300 rollen die rollen-kombinaties kiezen die een optimale vulling van de ovens verzekeren.

Gebruik makende van enkele vuistregels slaagt hij erin dit ingewikkelde probleem een redelijke oplossing te geven.

Onze taak was het bovengenoemde beslissingsname te automatiseren en te verbeteren. Gezien er geen algoritme voorhanden was, waren we aangewezen op heuristische programmatie. Het uitwerken van dit specifiek probleem heeft ons toegelaten enkele algemene principes van heuristische programmatie te toetsen.

## 2. Situering en werking van de gloeiafdeling in een geïntegreerd staalbedrijf.

De gloeiafdeling is gelegen in de koudwalserij, in lijn tussen de tandem-wals en de skin-pass (fig. 1 en 2).

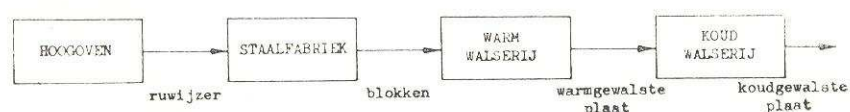


Fig. 1. — Geïntegreerd staalbedrijf.

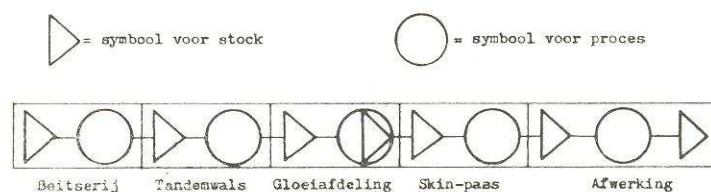


Fig. 2. — Koudwalserij.

Het gloeien is een herkristallisatieproces dat tot doel heeft de korrelgrootte te homogeniseren en aldus de nadelige gevolgen van het koudvervormen te doen verdwijnen, dit gebeurt in stolpovens met verplaatsbare klok.

De stolpoven heeft drie delen (fig. 3) :

— een vaste basis waarop de rollen worden gestapeld;

- een verplaatsbare beschermingsklok gevuld met inert gas om de oxydatie te vermijden;
- een verwarmingsstolp gestookt met gas.

De warmteoverdracht geschiedt voornamelijk door convectie, een ventilator opgesteld in de basis zorgt voor een degelijke stroming. Tussen de rollen worden convectoren geplaatst die deze convectiestroming nog intensifiëren.

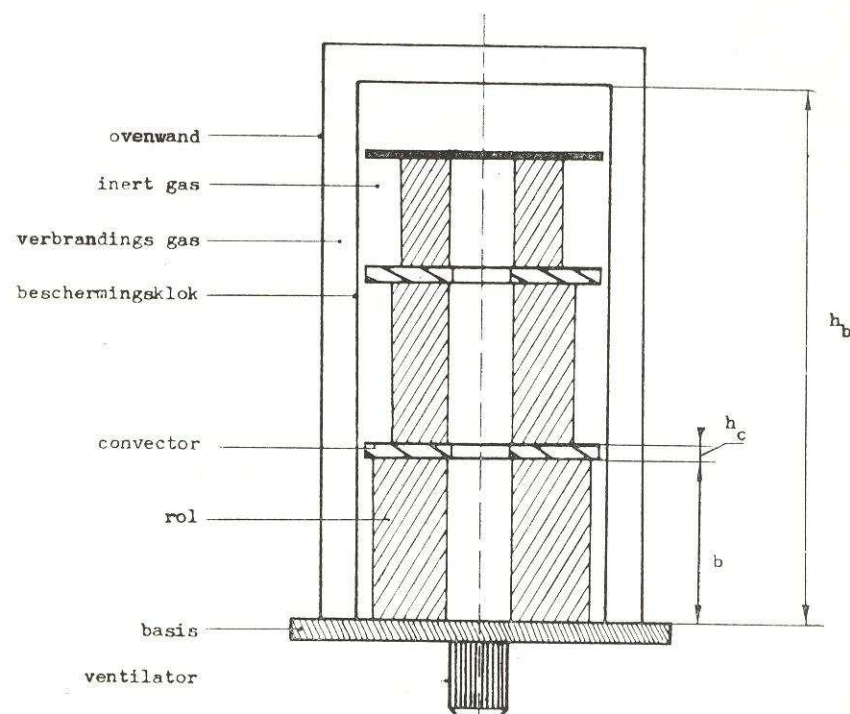


Fig. 3. — Gevulde oven.

De totale produktiecyclus bestaat uit :

- het laden van de rollen;
- het spoelen met inert gas (ongeveer 1 h.);
- het gloeien (gemiddeld ongeveer 22 h.);
- het afkoelen (na het gloeien wordt de verwarmingsstolp afgenomen en plaatst men de afkoelstolp waarin een ventilator ingebouwd is) (gemiddeld 35 h.);
- het ontladen.



Dagelijks kiest de verantwoordelijke uit de stock een aantal rollen die het dagelijks programma vormen. Deze rollen groepeer hij in stapels, waarbij hij tracht de stapelhoogte te maximaliseren.

| DIAM<br>KWAL. | $\phi \leq 1250$ | $1250 < \phi \leq 1500$ | $1500 < \phi \leq 1700$ | $\phi > 1700$ |
|---------------|------------------|-------------------------|-------------------------|---------------|
| CQ            |                  |                         |                         |               |
| DD            |                  |                         |                         |               |
| EDD           |                  |                         |                         |               |
| SEDD          |                  |                         |                         |               |
| SSEDD         |                  |                         |                         |               |

Fig. 4. — Subgroepen.

Om de gloeitijd echter zo laag mogelijk te houden (zie verder berekeningsmethode van de gloeitijd par. 3) beperkt hij zich zoveel mogelijk tot de rollen van een subgroep. Elke subgroep wordt gedefinieerd door een kwaliteit en een zone van de diameter (fig. 4).

Door deze dubbele verdeling in  $5 \times 4 = 20$  groepen bekomt men per subgroep een klein aantal rollen dat zich moeilijk tot optimale combinaties leent.

### 3. Berekeningsmethode van de gloeitijd.

Nemen we een heterogene stapel met 3 rollen, elke rol is gekarakteriseerd door een gewicht ( $G$ ), een dikte ( $d$ ), een breedte ( $b$ ), een dieptrek-kwaliteit ( $Q$ ) (\*) en een buitendiameter  $\phi$  (de binnendiameter is dezelfde voor alle rollen) (zie fig. 5).

Per rol wordt een rolgloeitijd uitgerekend, dit is de tijd die deze rol nodig heeft om goed gegloeid te worden. De gloeitijd is enkel een restrictie

(\*) Voor de dieptrekkwaliteit worden de volgende symbolen gebruik die in de E.G.K.S. gebruikelijk zijn :

|       |   |                                   |
|-------|---|-----------------------------------|
| CQ    | — | Commercial quality                |
| DD    | — | Deep drawing                      |
| EDD   | — | Extra deep drawing                |
| SEDD  | — | Special extra deep drawing        |
| SSEDD | — | Super special extra deep drawing. |

naar onder toe, wanneer de gloeitijd overschreden wordt verbetert de kwaliteit van de staalplaat.

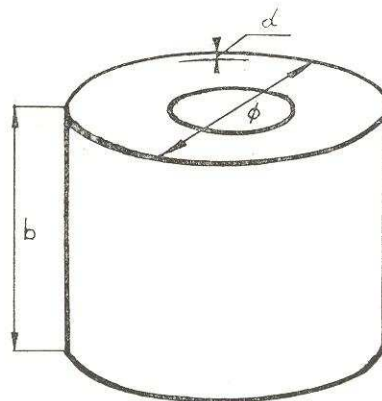


Fig. 5. — Rol.

De stapelgloeitijd is gelijk aan de grootste rolgloeitijd van de stapel. De rolgloeitijd is functie van de kwaliteit, het gewicht en de breedte van de rol en ook van de hoogte van de stapel ( $H$ ).

$$\text{Rolgloeitijd } T = T'(G_s, Q) \cdot B(b) \quad (\text{fig. 6})$$

$$G_s = \frac{G \cdot H}{b} \quad H = b_1 + b_2 + b_3$$

$G_s$  wordt het schijngewicht genoemd.

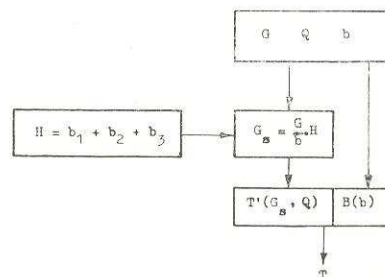


Fig. 6. — Berekening der rolgloeitijd.

$T'(G_s, Q)$  geeft een gloeitijd in functie van  $G_s$  en  $Q$ , (fig. 7) zonder rekening te houden met  $B(b)$ , een functie die deze gloeitijd verlengt of verkort naargelang de breedte (fig. 8).

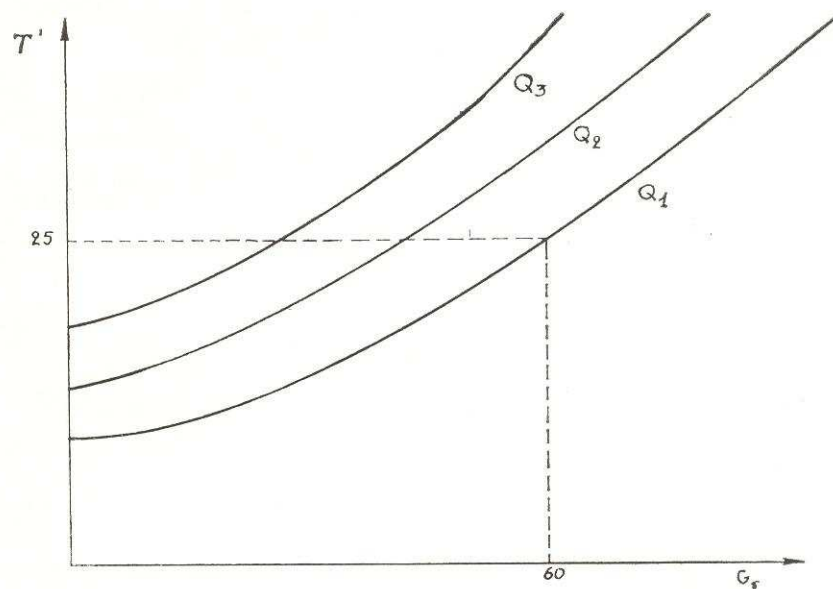
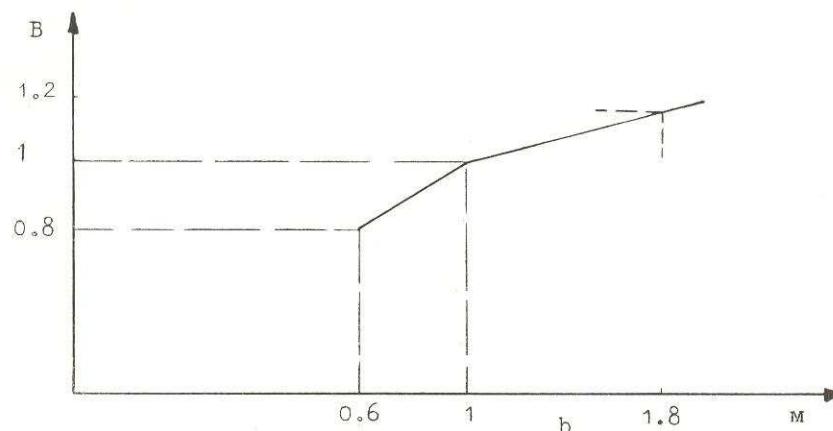


Fig. 7. — De gloeitijd in functie van schijngewicht en kwaliteit.

Fig. 8. — Korrektiefunctie  $B(b)$ .

Uit economisch standpunt is het niet interessant rollen met sterk verschillende gloeitijden te combineren in een stapel, daar zoals hoger vermeld de rol met de grootste gloeitijd determinerend is voor de stapelgloeitijd. Deze rolgloeitijden zijn echter slechts gekend nadat de stapel gevormd is (functie van stapelhoogte  $H$ ) en kunnen dus moeilijk als leidraad dienen bij het combineren.

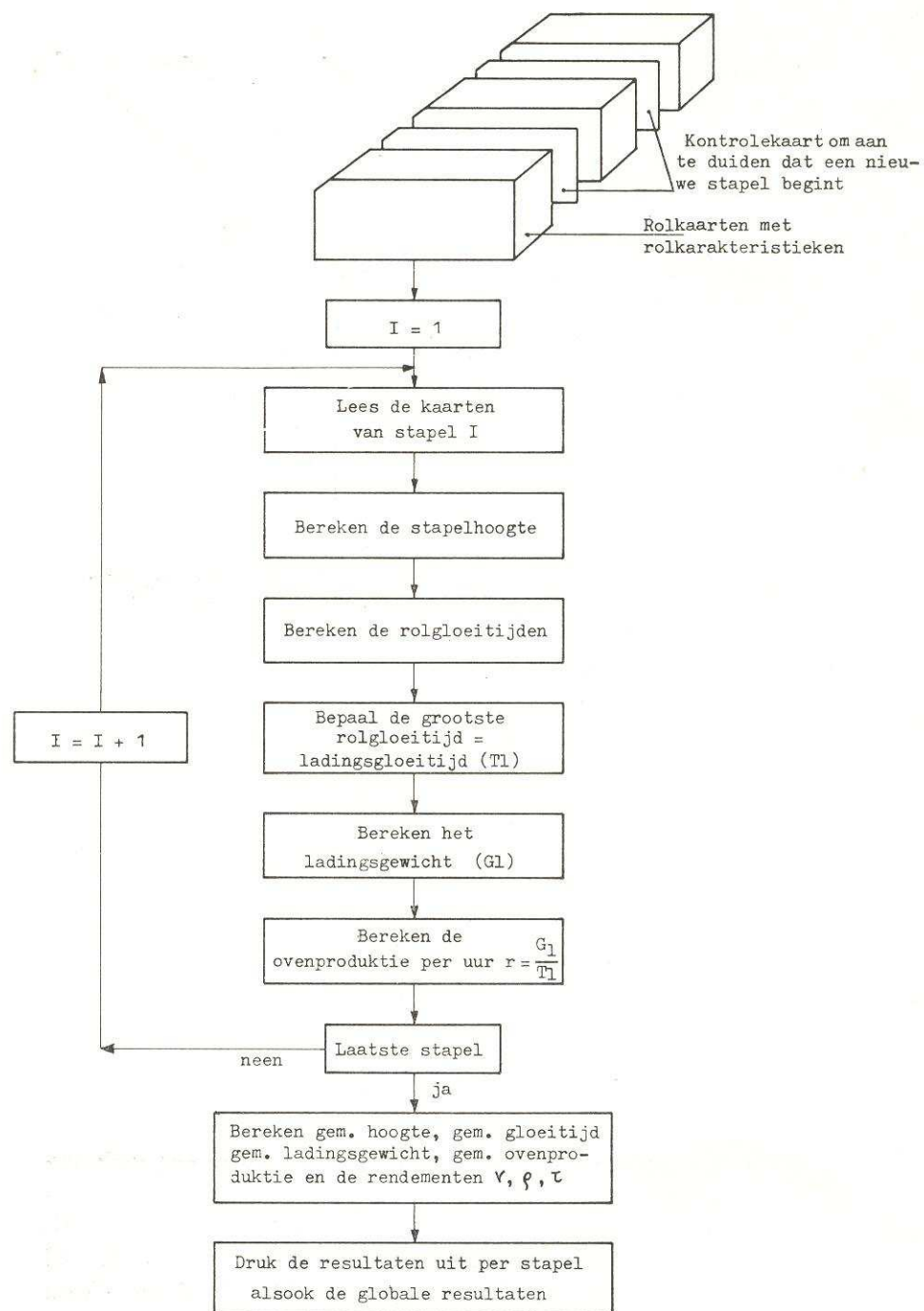


Fig. 9. — Programma A. Berekening der stapelgloeitijden.



De hierboven beschreven methode word omgezet in een computer-programma met als input een kaart per rol met de rolkaracteristieken en tussen de rolkaarten die samen horen in een stapel een controlekaart en als output een listing met de resultaten per stapel en de globale resultaten (fig. 9).

#### 4. Bespreking van de doelfunctie.

Zolang de gloeiafdeling het knelpunt niet is van het ganse productieproces moeten we de variabele kosten minimaliseren.

De voornaamste variabele kosten in de gloeiafdeling zijn de kosten van het verbrandingsgas, het inert gas en de electriciteit. Deze kosten zijn ongeveer konstant per lading, zodat de kosten per ton gedrukt worden door het gemiddeld stapelgewicht te verhogen. Daar de rolafmetingen als gegeven beschouwd worden kan dit stapelgewicht enkel verhoogd worden door de stapelhoogte te maximaliseren. Of wat op hetzelfde neerkomt voor een gegeven productieprogramma het aantal ladingen te minimaliseren.

Het vitaal produktiemiddel in de gloeiafdeling is de ovenstolp. Wanneer nu de gloeiafdeling het knelpunt is van de koudwalserij verandert de doelfunctie en moeten we de ovenproductie per uur maximaliseren.

$$\text{Gem. ovenproductie per uur} = \frac{\sum \text{ladingsgewichten}}{\sum \text{ladingsgloeitijden}} = \frac{G_t}{T_t} = \frac{G_{lm}}{T_{lm}} = r_m$$

$$G_{lm} = \text{gemiddeld ladingsgewicht} \quad T_{lm} = \text{gemiddelde ladingsgloeitijd}$$

Maar als we in onderbezetting werken moeten we er voor zorgen dat we niet in de knelpuntfase komen, dus mogen we de overproductie per uur niet uit het oog verliezen.

Dit is echter te realiseren door het feit dat een groot aantal oplossingen (= programma's) een goed gemiddeld ladingsgewicht opleveren, we moeten dus daaruit het programma kiezen dat terzelfdertijd een goede ovenproductie geeft.

De mathematische formulatie van de doelfunctie is zeer moeilijk. Met behulp echter van de heuristische programmatie kan een interessante oplossing gevonden worden zonder de doelfunctie mathematisch te formuleren, dit in tegenstelling met de mathematische programmatie.

#### 5. Maximalisatie van het gemiddeld ladingsgewicht.

In deze paragraaf stellen we ons enkel tot doel het gemiddeld ladingsgewicht te maximaliseren (aantal ladingen minimaliseren) en we verwaarlozen het aspekt gloeitijd.

De enige restrictie is dan dat de stapelhoogte kleiner moet zijn dan de ovenhoogte.

$b$  — breedte van de rol

$b_c$  — convectorhoogte

$b_b$  — beschikbare hoogte in beschermingsstolp

$N$  — aantal rollen (index rol =  $j$ )

$n'$  — aantal ovens ter beschikking (aantal vergelijkingen)

$n$  — aantal stapels (index stapel =  $i$ )

als  $x_{ij} = 1$  dan behoort rol  $j$  tot stapel  $i$

$x_{ij} = 0$  dan behoort rol  $j$  niet tot stapel  $i$

$n_i$  — aantal rollen in stapel  $i$

$$n_i = \sum_j^{1 \dots N} x_{ij}$$

Voeren we de veranderlijke  $\varepsilon_i$  in.

Als  $n_i > 0$  dan  $\varepsilon_i = 1$

$n_i = 0$  dan  $\varepsilon_i = 0$ .

De restrictievergelijkingen drukken uit dat de stapelhoogte kleiner is dan de beschikbare hoogte.

Dit geeft  $n'$  ongelijkheden van de vorm

$$\sum_j^{1 \dots N} x_{ij} (b_j + b_c) \leq b_b \quad (i = 1 \dots n')$$

Elke rol kan natuurlijk slechts eenmaal gestapeld worden, dit geeft  $N$  vergelijkingen van de vorm

$$\sum_i^{1 \dots n'} x_{ij} = 1 \quad (j = 1 \dots N)$$

De doelfunctie  $= n = \sum_i^{1 \dots N} \varepsilon_i$  moet geminimaliseerd worden.

In dit mathematisch model nemen de veranderlijken  $x_{ij}$  en  $\varepsilon_i$  de discrete waarden 0 of 1 aan.

We zijn dus verzeild in het domein van de integer programming waarin de veranderlijken binair zijn.

Deze problemen werden o.a. behandeld door GOMORY (Bibl. 2).

Voor  $n' = 40$  en  $N = 250$  zijn er dus 10.000  $x_{ij}$ .

## 6. Vooruitrekenen der rolgloetijden, maximalisatie van de ovenproduktie per uur.

De probleemstelling van paragraaf 5 is echter onvolledig daar de gloeitijd uit het oog werd verloren.

Het is echter niet mogelijk de gloeitijd te verwerken in de doelfunctie, daar de gloeitijd functie is van de stapelsamenstelling en de stapelsamenstelling de onbekende is van ons probleem.

We kunnen exhaustief het probleem oplossen door alle mogelijke programma's te onderzoeken en per programma het gemiddeld stapelgewicht en de gemiddelde gloeitijd te berekenen. Van zodra we echter een programma moeten opstellen voor een groot aantal rollen is dit zelfs met een moderne computer onbegonnen werk.

Het uittesten van combinaties mag echter niet verwaarloosd worden, daar de korte rekentijd van huidige computers toelaat in enkele sekonden 10.000 stapelcombinaties te genereren.

We moeten dus trachten uit het oneindig groot aantal combinaties, die combinaties te onderzoeken die een interessante oplossing kunnen geven.

De verantwoordelijke lost dit probleem op door een beperkte populatie te beschouwen, hij groepeerd namelijk die rollen die ongeveer eenzelfde gloeitijd hebben, daarbij baseert hij zich hoofdzakelijk op de elementen die de gloeitijd bepalen zoals kwaliteit, diameter, enz. (zie paragr. 1 - subgroepen), dit omdat hij de gloeitijd niet kent want de gloeitijd is ook afhankelijk van de stapelhoogte.

Indien we echter van de veronderstelling uitgaan dat we er in slagen de stapelhoogte te maximaliseren dan kunnen we aannemen dat de stapelhoogte niet veel zal afwijken van een gemiddelde stapelhoogte. In die veronderstelling zouden we de gloeitijd kunnen vooruitrekenen. Inderdaad alle elementen die deze gloeitijd bepalen zijn nu gekend  $G$ ,  $Q$ ,  $b$  en  $H = \text{gemiddelde stapelhoogte}$ . Deze vooruitgerekende gloeitijd zal zeker weinig verschillen van de werkelijke rolgloeitijd en kan dus als schatting dienen van de werkelijke rolgloeitijd en ook als richtlijn bij het combineren.

Aan de hand van bovenstaand idee werd een computerprogramma (zie fig. 10) opgebouwd met als input een kaart per rol met de rolkarakteristieken en als output een matrix  $R$  met de vooruitberekende rolgloetijden geklasseerd in dalende zin.

In de knelpuntfase, waar de nadruk ligt op het maximaliseren van de ovenproduktie, zou het probleem als volgt kunnen geformuleerd worden.

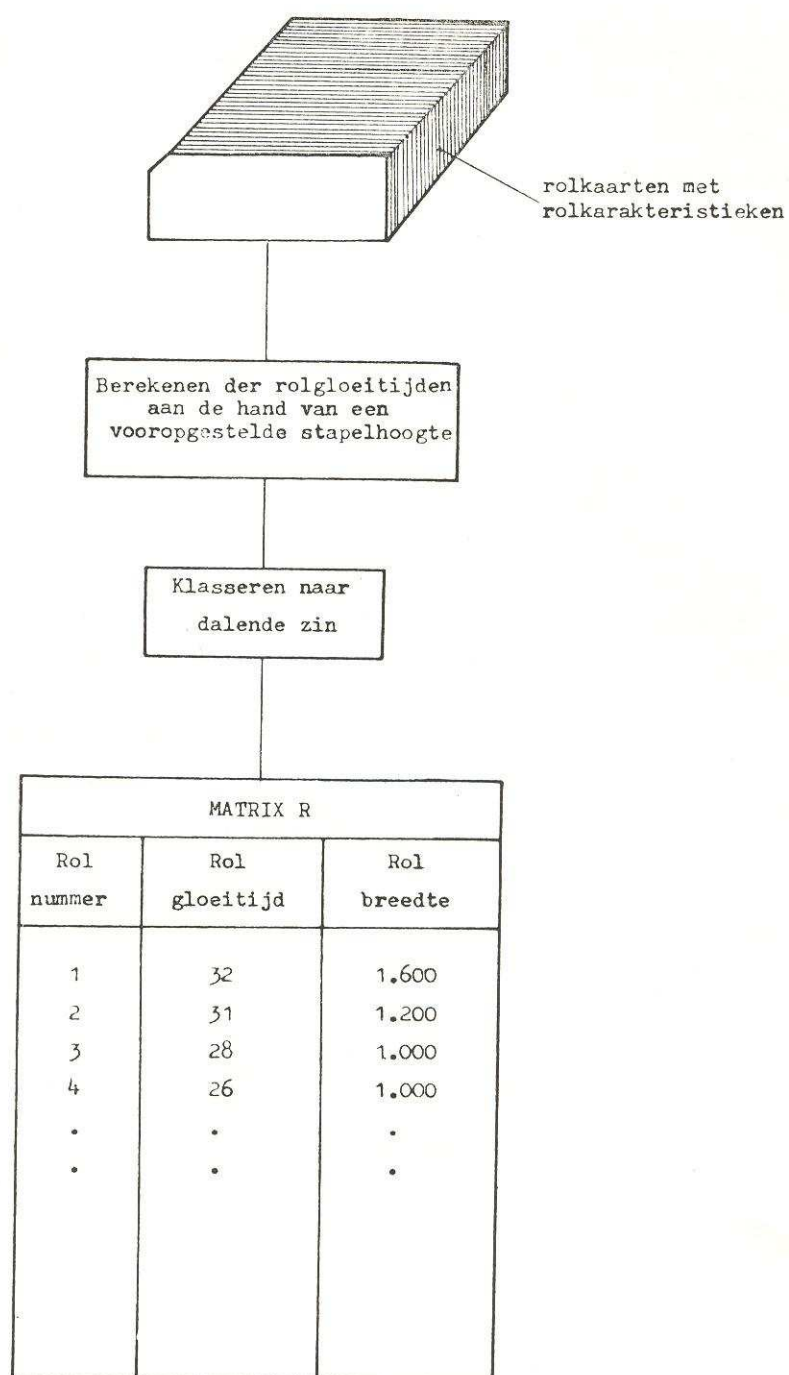


Fig. 10. — Programma B. Vooruitberekening der rolgloeitijden.



De restrikties zijn dezelfde als van het vereenvoudigd model (zie paragraaf 5). We moeten de doelfunctie  $r_m = \frac{\sum_{i=1}^n G_{li}}{\sum_{i=1}^n T_{li}}$  maximaliseren.

$\sum_{i=1}^n G_{li}$  is een konstante van een bepaald programma, nl. het totaal gewicht van het programma, dus  $r_m$  maximaliseren, komt neer op het minimaliseren van  $\sum_{i=1}^n T_{li} \approx \sum_{i=1}^N T_i \cdot \eta_i$ .

$\eta_i = 1$  of  $0$

$\eta_i = 1$  als rol  $i$  de rol is met de hoogste gloeitijd van de stapel

$\eta_i = 0$  als rol  $i$  niet de rol is met de hoogste gloeitijd van de stapel

$T_i$  = de vooruitberekende rolgloeitijd

$\sum_{i=1}^N \eta_i = n$  aantal stapels.

## 7. Objectieve vergelijkingspunten.

Een van de voorwaarden voor het al of niet aanwenden van heuristiek bij het oplossen van een probleem is het bestaan van objectieve vergelijkingspunten. Zoals vroeger vermeld zijn de twee voornaamste resultaten van een gloeiovenprogramma het gemiddeld stapelgewicht en de gemiddelde ovenproduktie per uur. Bij gelijkblijvende populaties zijn die resultaten ook een objectieve maat voor de efficiëntie van een heuristiek.

Bij wisselende populaties worden die resultaten vervalst door de karakteristieken van de populatie. Het is dan ook interessant die resultaten te ontleden en het verband te onderzoeken met de karakteristieken van de populatie. We mogen schrijven dat

$$G_{lm} \cdot n = G_{rm} \cdot N \quad G_{lm} = \frac{G_{rm} \cdot N}{n} \quad (1)$$

$G_{rm}$  = gemiddeld rolgewicht.

$G_{rm}$ ,  $N$  zijn onveranderlijke karakteristieken van de populatie.

Het aantal stapels  $n$  is een resultaat van de kombinatiopolitiek en van het minimum aantal stapels dat fysisch te realiseren is.

We kunnen  $v = \frac{n_{min}}{n}$  (2) als bezettingsrendement definiëren.

We berekenen nu  $n_{min}$ .

De nuttige hoogte van oven  $i$  is

$$b_{ni} = b_b - b_c \cdot n_i \quad (3)$$

Sommeren we over alle ladingen, dit geeft

$$b_{nt} = n \cdot b_b - b_c \cdot N$$

$$b_{nt} = \text{totale nuttige hoogte.}$$

Wanneer de totale nuttige hoogte bezet is vinden we een minimum aantal stapels.

$$\text{Dus } b_{nt} = \sum_{j=1}^{1-N} b_j = N \cdot b_m$$

$$b_m = \text{gemiddelde breedte}$$

$$\text{dit geeft } N \cdot b_m = n_{\min} \cdot b_b - b_c \cdot N$$

$$\text{of } n_{\min} = \frac{N \cdot (b_m + b_c)}{b_b} \quad (4)$$

We moeten  $n_{\min}$  afronden tot een geheel getal wil dit een fysische realiteit vertegenwoordigen.

$n_{\min}$  hangt dus van twee populatiekarakteristieken af, namelijk  $N$  en  $b_m$ ,  $b_c$  en  $b_b$  zijn constructiegegevens.

Substitueren we formule (2) en (4) in formule (1) dan bekomen we

$$G_{lm} = v \cdot G_{rm} \cdot \frac{b_b}{b_m + b_c} \quad (5)$$

$$\text{of nog } G_{lm} = v \cdot G_{lm \max} \quad (6) \quad G_{lm \max} = G_{rm} \cdot \frac{b_b}{b_m + b_c}$$

Bij gelijkblijvend bezettingsrendement oefent het gemiddeld rolgewicht en de gemiddelde breedte een grote invloed uit op het gemiddeld stapelgewicht. Het bezettingsrendement geeft ons een goede maat voor de efficiëntie van onze combinatiepolitiek qua stapelhoogte.

Om nu de efficiëntie van onze combinatiepolitiek te meten op de gemiddelde overproductie per uur gaan we over op de volgende formules

$$r_m = \frac{G_t}{T_t} = \frac{G_{lm}}{T_{lm}} \quad (7)$$

$G_{lm}$  hebben we reeds ontleed. Als minimum gemiddelde gloeitijd zouden we het gemiddelde van de vooruitgerekende gloeitijden kunnen nemen ( $T_{lm \min}$ ).

We definiëren (8)  $\tau = \frac{T_{lm \min}}{T_{lm}}$  als zijnde het gloeitijdsrendement.

Substitueren we formules (6) en (8) in formule (7), dit geeft

$$r_m = v \cdot \tau \frac{G_{lm \max}}{T_{lm \min}}$$

Of indien we  $\rho = v \cdot \tau$  en  $r_{m \max} = \frac{G_{lm \max}}{T_{lm \min}}$  stellen

$r_m = \rho \cdot r_{m \max}$ , noemen we het kombinatierendement

$$r_{m \max} = \frac{G_{mr} \cdot b_b}{(b_m + b_c) T_{lm \min}}$$

De minimum gemiddelde gloeitijd  $T_{lm \min}$  hangt natuurlijk ook af van de karakteristieken van de populatie en voornamelijk van de distributie van de kwaliteiten en van het gemiddeld rolgewicht. Approximatief kunnen we zeggen dat  $T_{lm \min} = g \cdot G_{mr} + b$  ( $g$  en  $b$  hangen af van de distributie van de kwaliteiten).

Het kombinatierendement  $\rho$  is een objectieve maat voor de combinatiepolitiek, zowel wat betreft vulling als gemiddelde gloeitijd.

Als we de studie begonnen hadden de rendementen de volgende waarden

$$v = 0,93 \quad \tau = 0,81 \quad \text{en} \quad \rho = 0,76.$$

## 8. Heuristische procedure.

### 81. Volgorde der bewerkingen.

811. Gezien er geen wiskundige algoritme bestond om het beschreven probleem op te lossen waren we aangewezen op heuristiek.

*De grondidee was de rollen met gelijke vooruitgerekende gloeitijd zoveel mogelijk te groeperen en in die groepering stapels te vormen met maximale hoogte.* (Heuristisch idee).

Met behulp van de listing (programma B) der vooruitgerekende gloeitijden die als leidraad dienen bij elke procedure, werd manueel een procedure uitgevoerd die voldeed aan bovenstaande heuristische regels (heuristische procedure).

Nadien werd het resultaat van die procedure berekend door programma A. Programma A geeft ons de resultaten per stapel zoals stapelhoogte, stapel-

gewicht, stapelgloeitijd en ovenproductie per uur, dit laat ons toe na te gaan voor welke stapels de procedure gefaald heeft.

Het programma geeft ons ook de globale resultaten die ons de mogelijkheid geven de procedure in zijn geheel te beoordelen.

De volgorde der bewerkingen wordt geschetst in figuur 11.

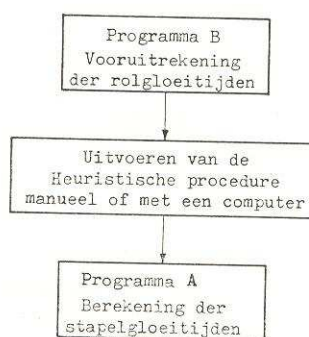


Fig. 11. — Volgorde der bewerkingen.

De volgende procedures werden manueel uitgevoerd.

812. Neem drie naast elkaar geschikte rollen in de rangschikking naar dalende gloeitijd en neem als vierde rol in de stapel die rol met de hoogste gloeitijd die in de oven kan.

In deze procedure ligt de nadruk op de gloeitijd en verwaarlozen we het aspekt stapelhoogte zodanig dat het gemiddeld stapelgewicht laag uitvalt, en we een relatief groot aantal stapels bekomen wat op zijn beurt de ovenproductie per uur drukt.

Deze procedure gaf dan ook resultaten die beneden de resultaten van de meestergasten lagen.

813. We hebben dan ook bovenstaande procedure verfijnd door een betere ovenvulling te viseren.

De procedure lijkt op de voorgaande maar nu neemt men als vierde rol, deze rol die de oven zo hoog mogelijk vult.

Deze procedure gaf opmerkelijk betere resultaten maar faalde toch dikwijls namelijk in die gevallen waar door de keus van de drie eerste rollen geen gunstige vierde te vinden was.

Ook gebeurde het dikwijls dat de vierde rol laag in de tijdschaal lag.



**82. Algemene voorwaarden waaraan een heuristische procedure moet voldoen.**

— Het resultaat van een goede procedure moet het optimum benaderen.

— Elke procedure moet voldoen aan de restrikties van het probleem. Het resultaat van gelijk welke procedure wordt natuurlijk ook bepaald door de populatie waarop de procedure wordt toegepast. Naarmate die populatie groter en heterogener is in de breedte zal het resultaat dan ook beter zijn.

Wij kunnen de basispopulatie vergroten door de stock voor gloeiovens te vergroten.

Hier moeten we het voordeel van een grote populatie afwegen tegen het nadeel dat een grote stock meebrengt.

Bovenstaande redenering leidt tot de volgende procedure-eisen :

— Een goede procedure vereist geen grote basispopulatie. Hoe zwakker een procedure, hoe groter basispopulatie ze vereist.

— Een procedure moet goede resultaten geven bij wisselende populaties. Deze voorwaarde is in het gloeiovenprobleem bijzonder van toepassing daar de produktieschakel vóór de gloeiafdeling, namelijk de tandemwals, oplegt dat in het werkprogramma de rollen geschikt moeten worden naar dalende rolbreedte.

**83. Het manueel uitvoeren van procedures.**

Het manueel uitvoeren van een procedure heeft de volgende voor- en nadelen.

— Wanneer we een heuristische procedure manueel uitvoeren dan kunnen we van dichtbij die procedure toetsen aan de populatie. Zo ervaren we wanneer de procedure faalt en in welke gevallen ze goede resultaten oplevert.

— Zolang een procedure manueel uitgevoerd wordt is ze zeer soepel daar ze gemakkelijk veranderd kan worden. Vooraleer een procedure te automatiseren is het dan ook geraadzaam ze manueel uit te voeren, op verscheidene populaties.

— Dit leert ons het probleem beter kennen en helpt ons de procedure te verfijnen en aan te passen aan wisselende populaties.

— Zolang een procedure manueel uitgevoerd wordt zijn we niet verplicht de procedure voor alle mogelijke gevallen te bepalen, we kunnen ons beperken tot de meest voorkomende gevallen en de anderen geval per geval oplossen.

— Een manueel uit te voeren procedure moet eenvoudig zijn daar een ingewikkelde procedure langdurig is.

— Zolang we een procedure manueel uitvoeren zoeken we de verbeteringen van die procedure in de richting en mogelijkheden beperkt door het manuele, m.a.w. men verliest de mogelijkheden van de computer uit het oog.

Het is dan ook geraadzaam na zekere testtijd de procedure te automatiseren en stelselmatig de geautomatiseerde procedure te verbeteren en te verfijnen. Volgorde 1 op figuur 12 is te verkiezen boven volgorde 2.

We hebben dan ook procedure 813 geprogrammeerd en het programma verder verfijnd.

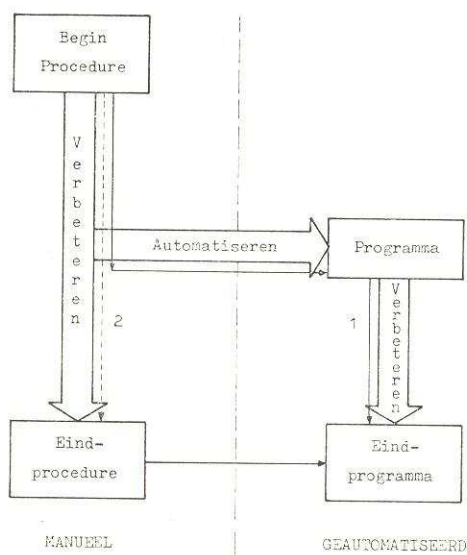


Fig. 12. — Verbetering van een beginprocedure.

## 9. Heuristische programmatie.

### 91. Het heuristisch programma.

Wanneer we een heuristische procedure automatiseren met behulp van een digitale computer spreken we van een heuristisch programma.

In tegenstelling met een manuele procedure moet een heuristisch programma streng opgesteld zijn. Alle gevallen moeten voorzien zijn en het programma moet goed functioneren bij wisselende populaties.

Dit stelt hoge eisen aan het programma dat moeilijk op punt te stellen is, de programmator moet een dieper inzicht in het probleem hebben.

Het opstellen van een heuristisch programma is dan ook een langdurig en kostelijk werk, dat veel computertesten vereist.

Deze kosten moeten afgewogen worden tegen de voordelen die het programma biedt.

Een heuristisch programma mag natuurlijk meer gecompliceerd zijn dan een manuele procedure. We mogen daarin echter niet overdrijven daar we bij grote populaties grote computertijden kunnen bekomen.

Wij hebben onze programma's geschreven in de symbolische taal Fortran IV.

Fortran is niet de enige taal die zich leent tot heuristische programmatie, er zijn andere talen zoals I.P.L. - V - Information Processing Language (Bibl. 3).

Gezien een heuristisch programma voortdurend gewijzigd wordt in de onderzoeksfase moet de symbolische taal waarin het geschreven wordt zeer soepel zijn.

Wijzigingen en aanvullingen in de procedure moeten kunnen vertaald worden in kleine wijzigingen in het programma, zodat we niet verplicht zijn gans het programma te herschrijven.

Naar onze ondervindingen leent Fortran zich zeer goed tot heuristische programmatie, een sterke verandering in een procedure had slechts de wijziging van een tiental instructies tot gevolg.

We moeten natuurlijk een verschil maken tussen het programma dat we gebruiken in de studiefase en het uitbatingsprogramma.

Het eerste programma wordt voortdurend gewijzigd en aangepast en het finale programma dat we zo bekomen is niet altijd efficiënt qua computertijd, de hoofdaandacht gaat immers naar het uitwerken van de heuristiek. Eer dat we dit programma in uitbating brengen gaan we het rationaliseren om de computertijd te verkorten, de inspanning voor rationalisatie is echter slechts lonend als dit programma frequent toegepast wordt.

#### 92. De Q-methode (fig. 13).

Een eerste verfijning van de geprogrammeerde manuele procedure was de volgende methode die we de Q-methode noemden.

Deze methode houdt rekening met de statistische verdeling van de breedte van de rollen. De gemiddelde breedte is een meter.

De methode is nu als volgt. Neem twee naast elkaar geschikte rollen in de rangschikking naar dalende gloeitijd. Bepaal de overblijvende plaats in de oven, dit is gemiddeld twee meter, deel die door twee. Dit quotient noemen we Q. Q is een getal dat rond een meter schommelt. Eerst onder-



zoeken we of deze  $Q$ -waarde kleiner of groter is dan de kleinste breedte (KLBR) in de populatie. Indien  $Q$  kleiner is dan kunnen er slechts drie rollen in de oven en nemen we als derde rol de grootste rol die nog in de oven kan. Indien  $Q$  groter is dan de kleinste dan onderzoeken we of er nog twee rollen in de oven kunnen, we doen dit door te onderzoeken of er in het gebied beperkt door de kleinste breedte en  $2Q$ -kleinste breedte tenminste twee rollen aanwezig zijn. Indien dit het geval is nemen we als derde rol een rol die  $Q$  in breedte zo dicht mogelijk benadert; daar  $Q$  nu een meter benadert is er een grote kans dat we symmetrisch van de  $Q$ -as een vierde rol zullen vinden die de oven maximaal vult.

Deze methode gaf reeds betere resultaten dan de vorige.

Zij is echter nog afgestemd op het manueel uitvoeren en gebruikt de mogelijkheden van de computer niet.

Dit programma is afgestemd op stapels gevormd uit vier rollen, d.i. het meest voorkomende, voor stapels gevormd uit drie of vijf rollen kan de heuristiek wel een goede stapel bekomen maar dit is zeker niet gegarandeerd.

Bovenstaande heuristische regels geven in de meeste gevallen goede stapelhoogtes, nochtans is dit niet verzekerd en gebeurt het dat we een lage stapelhoogte bekomen, in dit geval gaan we een andere combinatie beginrollen kiezen en de cyclus opnieuw beginnen (fig. 14).

We doen dit door te onderzoeken of dat de stapelhoogte een bepaalde limiethoogte ( $H_L$ ) overschreden heeft, is dit niet het geval dan herbegint de cyclus en nemen we als beginrollen de rollen 2 en 3 i.p.v. 1 en 2.

De procedure om lage stapelhoogte te vermijden werd naderhand bijgevoegd en dit leverde wegens de structuur van het programma geen enkele moeilijkheid op. (Vooral nuttig op het einde van een populatie).

Bij zeer slechte populaties leidt bovenstaande verwerpingsprocedure tot zeer veel recyclages en dit meestal zonder veel succes. Daarom wordt het aantal recyclages gelimiteerd, van zodra de limiet (KL) bereikt wordt stopt het programma en worden de resterende rollen van matrix  $R$  uitgedrukt.

Wanneer we voor een kleine populatie  $n_{\min}$  berekenen zonder af te ronden vinden we dikwijls een getal dat zeer dicht ligt bij het geheel getal dat juist kleiner is - bijv. 10,1 ligt zeer dicht bij 10.

Dit betekent dat als we de populatie behouden zoals ze is, we minimum 11 stapels kunnen bekomen, met een vrije hoogte van  $0,9 h_b$ . Dit is een resultaat dat we gemakkelijk zonder computer kunnen bekomen. Bij zulk een populatie kunnen we slechts een goede oplossing bekomen als we er



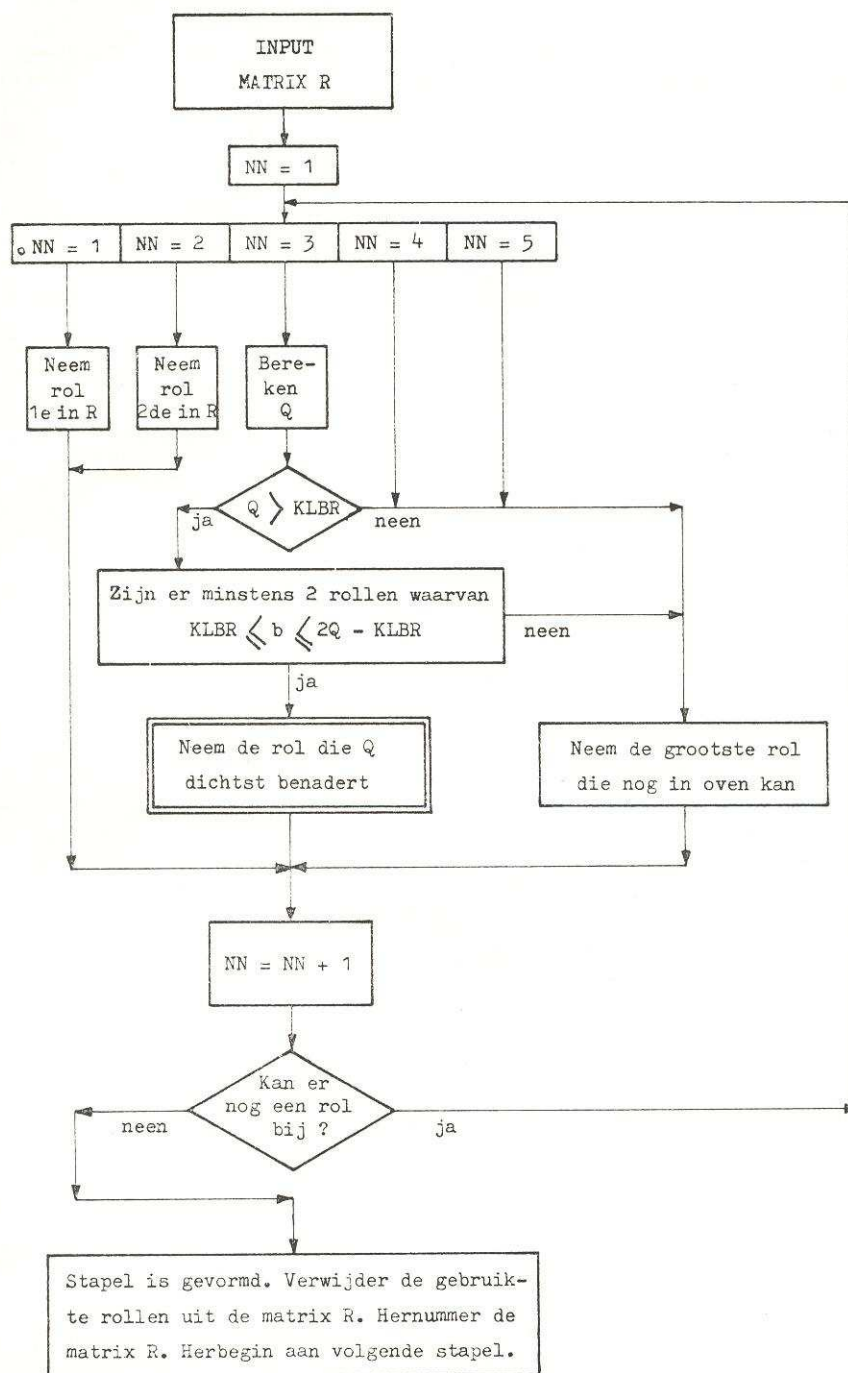


Fig. 13. — Q-Methode.

rollen uit verwijderen zodanig dat  $n_{\min}$  het juist groter zijnde geheel getal zeer dicht benadert.

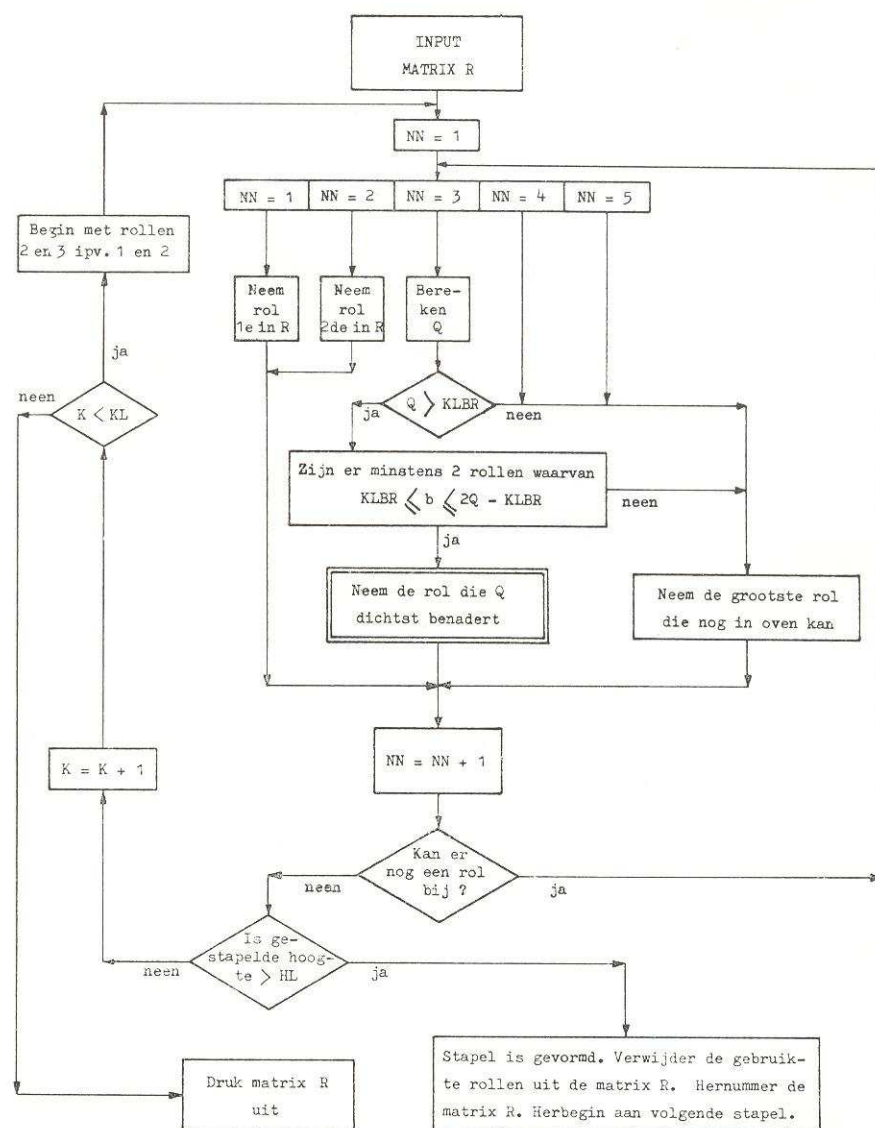


Fig. 14. — Q-Methode met controle op de stapelhoogte.

Dit verwijderen gebeurt op het einde van een populatie automatisch door de verwerpsprocedure.

**93. De kombinatorische methode.**

Overstappen van de Q-methode naar een meer geraffineerde methode leverde weinig moeilijkheden op wat betreft de programma-aanpassing.

Daartoe werd het deel van het organigram (fig. 15) dat dubbel omlijnd is vervangen door de volgende instructies.

— Vorm de combinaties (3,4) (3,5) ..... (3,N)  
                                   (4,5) (4,6) ..... (4,N)  
                                   (N—1, N)

tot wanneer een combinatie  $j, k$  voldoet aan de voorwaarde

$$2Q - b_j - b_k < \varepsilon$$

- Indien geen enkele combinatie voldoet aan die voorwaarde, neem deze combinatie die de oven maximaal vult.
- Neem rol  $j$  en dan rol  $k$ .
- $NN = NN + 1$ .

Een groot aantal combinaties valt a priori weg omdat de stapelhoogte de beschikbare hoogte overschrijdt.

Door de combinaties in de opgegeven volgorde te vormen elimineren we vlug de rollen met hoge gloeitijd in matrix R en bekomen we een lage gemiddelde gloeitijd. Hoe groter  $\varepsilon$  hoe vlugger we in de hoge gloeitijden een combinatie vinden die aan bovenstaande ongelijkheid voldoet, dit betekent winst op de gemiddelde gloeitijd en winst op de computertijd.

Deze dubbele winst wordt echter ingeboet door een systematisch verlies op de stapelhoogte.

We vinden een compromis door een juiste keuze van  $\varepsilon$ .

Bovenstaand programma gaf uitstekende resultaten. Voor  $v$ ,  $\tau$  en  $\rho$  bekwamen we respectievelijk 0,99, 0,89 en 0,88 — dit betekent een grote winst t.o.v. de handmethode ( $v = 0,93$ ,  $\tau = 0,81$  en  $\rho = 0,76$ ).

Het programma heeft 3 parameters nl. HL,  $\varepsilon$  en KL, deze parameters kunnen zonder enige moeilijkheid gewijzigd worden.

De keuze van HL,  $\varepsilon$  en KL is zeer delikaat en bepaalt de rentabiliteit van een programma.

In Sidmar werden deze parameters proefondervindelijk bepaald, t.t.z. voor een bepaalde populatie werd het verband bestudeerd tussen  $v$  en  $\tau$  en de verschillende parameters. Dit vereiste natuurlijk veel computerruns.

### 10. Besluit.

Veel mathematici staan sceptisch t.o.v. de heuristische programmatie, nochtans is dit een noodzaak geworden voor belangrijke problemen waarvoor geen algorithmische bestaat.

Weliswaar bereikt de heuristische programmatie het optimum niet maar benadert dit toch dicht.

Gebruikmakende van de menselijke feeling en de mogelijkheden van de computer, en door samenspraak van mens en computer, worden resultaten bereikt die voorheen onmogelijk waren.

Daar waar de klassieke operationele technieken falen biedt de heuristische programmatie een zeer interessante oplossing aan; als zodanig vult zij een leemte in huidige beslissingstechnieken.

Naar onze mening ligt voor de heuristische programmatie een ruim toepassingsveld open.

### Annexe A : Symbolen.

|          |   |                                                                   |
|----------|---|-------------------------------------------------------------------|
| $b$      | : | breedte van de rol                                                |
| $d$      | : | dikte van de rol                                                  |
| $\phi$   | : | diameter van de rol                                               |
| $G$      | : | gewicht van de rol                                                |
| $Q$      | : | kwaliteit van de rol                                              |
| $H$      | : | stapelhoogte (zonder convectoren)                                 |
| $G_l$    | : | ladingsgewicht                                                    |
| $G_{lm}$ | : | gemiddeld ladingsgewicht                                          |
| $G_r$    | : | rolgewicht                                                        |
| $G_{rm}$ | : | gemiddeld rolgewicht                                              |
| $G_t$    | : | totaal programma gewicht                                          |
| $T_l$    | : | ladingsgloeitijd                                                  |
| $T_{lm}$ | : | gemiddelde ladingsgloeitijd                                       |
| $T_t$    | : | totale programmagloeitijd                                         |
| $r$      | : | ovenproductie per uur                                             |
| $r_m$    | : | gemiddelde ovenproductie of ovenproductie per uur over alle ovens |
| $n$      | : | aantal stapels                                                    |
| $N$      | : | aantal rollen                                                     |
| $b_c$    | : | convectorhoogte                                                   |
| $b_b$    | : | beschikbare hoogte in de beschermingsstolp                        |



|        |   |                     |
|--------|---|---------------------|
| $b_n$  | : | nuttige hoogte      |
| $v$    | : | bezettingsrendement |
| $\tau$ | : | gloeitijdsrendement |
| $\rho$ | : | kombinatierendement |
| KL     | : | aantal recyclages   |
| HL     | : | limiethoogte        |
| KLBR   | : | kleinste breedte.   |

### Annexe B : Bibliografie.

- [1] Karg R.L., Thompson G.L., " A heuristic approach to solving travelling salesman problems ", Management Science, Vol. 10, N° 2, January 1964.
- [2] Gomory R., " Essentials of an algorithm for Integer Solutions to Linear Programs ", Bulletin of the American Mathematical Society (April 23, 1958).
- [3] Newell A., Tonge F.M., " An introduction to information processing language V - Communications of the association for computing machinery " (April 1960).
- [4] Newell A., Shaw J.C. and Simon H.A., " Empirical Explorations of the Logic Theory Machine : A case Study of Heuristics ", Proceedings of the Western Joint Computer Conference, Ire, February 1957.
- [5] Simon H.A. and Newell A., " Heuristic Problem Solving : The next advance in operations research ", operations research, Vol. 6, N° 1 (January-February 1958).
- [6] Newell A., Shaw J.C. and Simon H.A., " Elements of a Theory of Human Problem Solving ", Psychological Review, 65 : 151-166, May 1958.
- [7] Newell A., Shaw J.C. and Simon H.A., " Report on a General Problem-Solving Program ", Proceedings of the International Conference on Information Processing, UNESCO, Paris, 1959.
- [8] Clarkson G.P.E., " Portfolio Selection a Simulation of Trust Investment ", Prentice Hall inc. 1962, Englewood Cliffs, N.J.  
The Ford Foundation doctoral dissertation winners.
- [9] Tonge F.M., " A heuristic program for assembly line Balancing ", Prentice Hall inc. 1961, Englewood Cliffs, N.J., The Ford foundation doctoral dissertation series.
- [10] Lloyd M. Thayer, " Heuristic programming techniques in decision making systems ", Data processing yearbook.
- [11] Kuchn A.A. and Hamburger M.J., " Heuristic program for locating Warehouses ", Management Science, N° 9, July 1963.
- [12] Gavett J. William, " Three heuristic rules for sequencing jobs to a single production facility ", Management Science 11, N° 8, June 1965.
- [13] Gere Williams, " Heuristic Job Shop Scheduling », Management Science 13, N° 3, November 1966.
- [14] Hurst Gerald and Mc. Namara Aidan B., " Heuristic Scheduling in a woollen mill ", Management Science 14, N° 4, December 1967.