

# Optimalisatie van het snijproces van patronen in de textielindustrie

Annemieke van Dongen  
Stageverslag  
Voorjaar 2006

**vrije Universiteit**  
Faculteit der Exacte Wetenschappen  
Bedrijfswiskunde en Informatica  
De Boelelaan 1083  
1081 HV Amsterdam  
**Stagebedrijf:**  
Beach Life Fashion BV  
Einsteinweg 33e  
3752 LW Bunschoten



## Voorwoord

Dit verslag vormt de afsluiting van mijn stage bij Beach Life die ik in januari 2006 begon. Deze stage is het afsluitende onderdeel van de studie Bedrijfskunde en Informatica aan de Vrije Universiteit te Amsterdam.

Allereerst wil ik graag Beach Life bedanken dat ik de mogelijkheid heb gekregen om daar stage te lopen en voor de interessante stageopdracht.

Verder wil ik ook nog een aantal mensen bedanken:

- GJ Franx, die gedurende het eerste deel van mijn stage mijn begeleider vanuit de Vrije Universiteit was. Ik wil hem graag bedanken voor de ideeën en begeleiding die hij mij gegeven heeft.
- Evert Wattel, die tijdens het laatste deel van mijn stage mijn begeleider vanuit de VU was. Ook hem wil ik bedanken voor zijn begeleiding.
- Frank ter Horst, mijn begeleider bij Beach Life. Hem wil ik bedanken voor de samenwerking en alle informatie die hij mij gegeven heeft.
- Gerrit Timmer, de tweede lezer van mijn stage, wil ik graag bedanken voor zijn advies en voor het zijn van de tweede lezer.

Ook wil ik alle andere mensen die mij geholpen of gesteund hebben tijdens mijn stage bedanken.

Annemieke van Dongen  
Zaandam, juli 2006



## Samenvatting

Dit is het stageverslag van mijn stage bij Beach Life. Beach Life is het grootste trendy en modieuze badmodemerken in Nederland.

Om het proces van het snijden van de stof te plannen moeten intekeningen gemaakt worden. De opdracht van deze stage is het vaststellen van een optimalisatie-methode voor het bepalen van intekeningen, die rekening houdt met alle beperkende voorwaarden, zodat het proces van snijden kan worden geoptimaliseerd.

Dit probleem bleek erg moeilijk te zijn en is het is waarschijnlijk onmogelijk om dit probleem optimaal op te lossen. Daarom is besloten om dit probleem heuristisch aan te pakken.

Er zijn verschillende heuristieken ontwikkeld, zowel constructieheuristieken als verbeteringsheuristieken. Deze heuristieken maken het mogelijk om verschillende cupmaten en werkorders met elkaar te combineren. Ook bieden zij de mogelijkheid om verschillende kleuren stof tegelijkertijd te snijden.

De heuristieken zijn geïmplementeerd en getest. De resultaten zijn goed, de door het programma gegenereerde oplossingen scoren beter op alle criteria dan de oude oplossingen, of de met de hand gemaakte oplossingen.

Conclusie is dat de heuristieken erg goed werken. De beste oplossingen worden gevonden wanneer zo veel mogelijk werkorders en kleuren gecombineerd worden.

Het is echter niet altijd handig om alles te combineren, want dan wordt het overzicht over het productieproces snel verloren, hier dient dus een middenweg gevonden te worden.



# Inhoudsopgave

<b>VOORWOORD</b> .....	<b>3</b>
<b>SAMENVATTING</b> .....	<b>5</b>
<b>INHOUDSOPGAVE</b> .....	<b>7</b>
<b>HOOFDSTUK 1 INLEIDING</b> .....	<b>9</b>
BEACH LIFE .....	9
PROBLEEMSTELLING .....	9
STRUCTUUR VAN HET VERSLAG .....	10
<b>HOOFDSTUK 2 MODEL/PROCESBESCHRIJVING</b> .....	<b>11</b>
HUIDIGE PROCES VAN SNIJDEN .....	11
PUNTEN VOOR VERBETERING .....	13
<i>Verbeteren van de planning van het snijden</i> .....	13
<i>Stofefficiëntie</i> .....	14
CRITERIUMFUNCTIE .....	14
<i>De productietijd</i> .....	14
<i>De hoeveelheid verbruikte stof</i> .....	14
<i>Overproductie</i> .....	15
<i>Samenvatting criteriumfunctie</i> .....	15
WISKUNDIG MODEL .....	15
<i>IP model</i> .....	16
<i>Complexiteit</i> .....	17
<b>HOOFDSTUK 3 HEURISTIEKEN</b> .....	<b>19</b>
EÉN KLEUR.....	19
<i>Constructieheuristiek</i> .....	19
<i>Verbeteringsheuristiek 1: ander lagenaantal</i> .....	21
<i>Verbeteringsheuristiek 2: overproductie</i> .....	26
MEERDERE KLEUREN .....	27
<i>Constructieheuristiek</i> .....	27
<i>Verbeteringsheuristiek 1: andere lagencombinatie</i> .....	29
<i>Verbeteringsheuristiek 2: overproductie</i> .....	31
HET BEPALEN VAN DE BESTE OPLOSSING .....	33
<i>Stofefficiëntie</i> .....	34
ZWAKKE PUNTEN VAN DE HEURISTIEKEN.....	34
<b>HOOFDSTUK 4 IMPLEMENTATIE EN TESTS</b> .....	<b>37</b>
HET PROGRAMMA .....	37
TESTRESULTATEN .....	41
<i>Werkorder 5122</i> .....	41
<i>Werkorder 4163</i> .....	44
<i>Werkorders 5102 en 5158</i> .....	44
<i>De snelheid van het programma</i> .....	45
<i>Conclusies</i> .....	45
MEERDERE STOFSOORTEN .....	45
<b>HOOFDSTUK 5 CONCLUSIES EN AANBEVELINGEN</b> .....	<b>47</b>
CONCLUSIES.....	47
AANBEVELINGEN .....	47
<b>BIJLAGE A: NESTERONDERZOEK</b> .....	<b>49</b>
INLEIDING .....	49
ONDERZOEK 1 .....	49
<i>Onderzoeksvraag</i> .....	49
<i>Werkwijze</i> .....	49
<i>Resultaat</i> .....	49

<i>Conclusie onderzoek 1</i> .....	51
ONDERZOEK 2.....	51
<i>Onderzoeksvraag</i> .....	51
<i>Werkwijze</i> .....	51
<i>Resultaat</i> .....	51
<i>Conclusie onderzoek 2</i> .....	52
EXTRA ONDERZOEK.....	52
EINDCONCLUSIES NESTERONDERZOEK.....	53
<b>BIJLAGE B: SNIJLIJSTEN VAN GETESTE WERKORDERS</b> .....	<b>54</b>
WERKORDER 4163.....	54
WERKORDERS 5152 EN 5158.....	55



# Hoofdstuk 1 Inleiding

## **Beach Life**

Beach Life is het grootste trendy en modieuze badmodemerk in Nederland en is zowel actief in de sport- als in de body fashion branche. Ook internationaal gezien verwerft Beach Life steeds meer aanzien. Gebruik wordt gemaakt van hoogwaardige materialen die grotendeels van Italiaanse origine zijn, in combinatie met de unieke styling heeft de Beach Life collectie een uitermate onderscheidend en innovatief karakter.

Beach Life is een jonge en dynamische organisatie waarin 20 gemotiveerde mensen werkzaam zijn. Het hele proces van het ontwerpen (stylen), verkoop, productie en distributie vindt in eigen beheer plaats.

Op de styling afdeling worden allereerst ontwerpen gemaakt. Hierbij wordt rekening gehouden met de wensen van de doelgroep. Van deze ontwerpen worden proefproducten gemaakt. Uiteindelijk ontstaat zo een productlijn.

De volgende stap is de verkoop van de producten. Klanten kunnen verschillende producten en maten bestellen. Deze bestellingen resulteren in verschillende werkorders.

Naar aanleiding van deze werkorders wordt de stof ingekocht. De stof wordt opgestuurd naar de plaats van productie. Daar wordt de stof gesneden en worden de producten in elkaar gezet.

Na de productie worden de producten uitgeleverd aan de klanten. Daarna kan het hele proces weer overnieuw beginnen.

## **Probleemstelling**

Een van de bezigheden van Beach Life is de productie van de bestelde producten. Bij de productie is het van belang dat het snijden van de stof zo snel (efficiënt) mogelijk kan verlopen en dat daarbij zo min mogelijk stof verspild wordt. Om snel te kunnen snijden is het handig om door meerdere lagen stof tegelijk te snijden, zodat in één snijbeweging meerdere producten uitgesneden worden.

Om het proces van snijden te plannen moeten **intekeningen** gemaakt worden. Een intekening is een tekening van een aantal patroondelen die uit een bepaalde stof gesneden moeten worden. Een patroondeel is een deel van een product, bijvoorbeeld de voorkant van een bikinibroekje. Aan de hand van deze intekeningen worden de stoffen gesneden. In een intekening kunnen patroondelen van meerdere producten zitten en van een intekening kunnen meerdere lagen tegelijkertijd gesneden worden.

Om het snijproces te optimaliseren is het van wezenlijk belang dat men met zo min mogelijk intekeningen werkt, en bij elke intekening zo veel mogelijk lagen stof tegelijkertijd snijdt. Echter hierbij dient rekening te worden gehouden met een groot aantal factoren, zoals het aantal producten dat nodig is van elke maat en elk model, de stofbreedte, het maximaal aantal stoflagen, de lengte van de intekentafel etc.

De opdracht van deze stage is het vaststellen van een optimalisatie-methode voor het bepalen van de intekeningen, die rekening houdt met alle beperkende voorwaarden, zodat het proces van intekenen kan worden geoptimaliseerd. Tevens dient deze optimalisatie-methode geïmplementeerd te worden in de bestaande software, opdat het hele proces van intekenen volledig is geautomatiseerd.

## ***Structuur van het verslag***

Dit hoofdstuk is een inleiding en bevat een korte omschrijving van het bedrijf Beach Life. Ook bevat dit hoofdstuk de probleemstelling.

Het tweede hoofdstuk is de procesbeschrijving. Eerst wordt beschreven hoe de huidige gang van zaken is omtrent het maken van intekeningen en snijden van patronen. Vervolgens wordt beschreven waar de knelpunten zitten en waar ruimte voor verbetering ligt. Dan wordt de criteriumfunctie beschreven. Als laatste wordt een wiskundig model van het inteken-proces gegeven.

Het derde hoofdstuk gaat over de heuristieken die gemaakt zijn om dit probleem op te lossen. Er wordt van alle heuristieken een stappenplan gegeven en een uitleg. Ook wordt uitgelegd hoe de beste oplossing gekozen kan worden en wat de zwakkere punten van de heuristieken zijn.

In het vierde hoofdstuk staat hoe de heuristieken zijn geïmplementeerd en hoe ze zijn getest. Ook worden de testresultaten gegeven in dit hoofdstuk.

Het laatste hoofdstuk bevat mijn conclusies en aanbevelingen.

## Hoofdstuk 2 Model/procesbeschrijving

Dit hoofdstuk bevat de procesbeschrijving. Eerst wordt beschreven hoe de huidige gang van zaken is omtrent het maken van de tekeningen en het snijden van de patronen. Er worden tevens een aantal begrippen uitgelegd. In de tweede paragraaf wordt beschreven waar ruimte voor verbetering ligt. In de derde paragraaf wordt beschreven welke criteria belangrijk zijn voor de kwaliteit van een oplossing. Dit resulteert in de criteriumfunctie. Als laatste wordt een wiskundig model van het inteken-proces, en de complexiteit daarvan beschreven.

### **Huidige proces van snijden**

In het computersysteem staan alle *werkorders* die geproduceerd moeten worden.

#### **Werkorder:**

Op een werkorder staat hoeveel van een product moet worden geproduceerd van welke maten en kleuren.

Bijvoorbeeld:

Kleur	Cupmaat	Maat 38	Maat 40	Maat 42	Maat 44	Maat 46	Totaal
601	B	26	32	28	8	4	98
601	C	20	40	40	28	8	136
801	B	12	14	12	6	4	48
801	C	14	18	20	14	1	67
Totaal		72	104	100	56	17	349

#### **Product, item, patroondeel:**

Een product is een bepaald model bikini, badpak, zwembroek, etc. Bijvoorbeeld badpak type 5555 of bikini type 680.

Een item is een bepaald product van een bepaalde maat, bijvoorbeeld: badpak type 5555 maat 38 of bikini type 680, maat 40C.

Een patroondeel is een deel van een product. Een bikini bestaat bijvoorbeeld uit de voorkant van het broekje, de achterkant van het broekje, de linkercup van het bovendeel, etc.

Bij elk product hoort een *PISformulier*.

#### **PISformulier:**

Op een PISformulier van een product staat vermeld uit welke onderdelen dit product allemaal bestaat: patroondelen, (prijz)kaartjes, voering, stickers etc. Ook staat hierop vermeld hoeveel stof dit product gemiddeld kost.

Uit die werkorders en PISformulieren is precies bekend hoeveel er van alle patroondelen uitgesneden moet worden. Per werkorder moet een *snijlijst* gemaakt worden die aan de behoefte van de werkorder voldoet.

#### **Snijlijst:**

Een snijlijst geeft aan hoe het snijden gepland moet worden. Elke regel staat symbool voor een *intekening*; er staat op deze regel voor elk item hoeveel van dit item in de intekening moet komen. Ook staat er op deze regel hoeveel lagen stof opgelegd moet worden bij deze intekening.

Bijvoorbeeld:

Intekening	Kleur	Cup	38	40	42	44	46	Lagen
1	601	B	2	2	2	0	0	13
2	601	B	0	3	1	4	2	2
3	601	C	2	4	4	2	0	10
4	601	C	0	0	0	4	4	2

**Intekening:**

Een intekening is een tekening van een aantal patroondelen die uit een bepaalde stof gesneden moeten worden.

Uit een snijlijst kan worden terugberekend hoeveel van elk item wordt geproduceerd. Dit kan worden gedaan door het aantal lagen te vermenigvuldigen met het aantal maal dat een item in een intekening staat en dit voor alle intekeningen op te tellen. In het geval van het voorbeeld:

Kleur	Cupmaat	Maat 38	Maat 40	Maat 42	Maat 44	Maat 46	Totaal
601	B	26	32	28	8	4	98
601	C	20	40	40	28	8	136

Voor maat 38B is het getal 26 op de volgende manier berekend:  $2 \cdot 13 + 0 \cdot 2 = 26$ ,

voor maat 40B is het getal 32 op de volgende manier berekend:  $2 \cdot 13 + 3 \cdot 2 = 32$ .

Deze aantallen moeten hetzelfde zijn als de aantallen op de werkorder. Het is soms ook mogelijk dat er iets meer geproduceerd wordt dan dat de vraag is. Dit wordt *overproductie* genoemd.

Als een snijlijst gemaakt is, worden de daadwerkelijke intekeningen gemaakt. Dit gebeurt met de *nester*.

**Nester:**

De nester is een computerprogramma dat patroondelen efficiënt in elkaar legt. Input hiervoor is:

- de stofbreedte,
- het aantal stuks wat van elk item nodig is,
- de vorm van de patroondelen van de items,
- krimpprocentage,
- de stofrichting.

Uit dit programma rolt een plaatje hoe de patroondelen mooi in elkaar gelegd kunnen worden (een intekening).

Dit programma heeft wel enige tijd nodig om dit te berekenen, hiervoor wordt maximaal 7 minuten uitgetrokken.

Van de snijlijst worden alle regels een voor een ingevoerd in de nester. Van elke regel maakt de nester dan een intekening.

De benodigde stof is 4 cm langer dan de door de nester berekende intekening, omdat aan beide kanten 2 cm extra stof genomen wordt, omdat je niet helemaal tot de rand van de stof kunt snijden.

Dit aantal meter wordt weer ingevuld op de snijlijst, bij de betreffende intekening. Dit aantal, vermenigvuldigd met het aantal lagen vormt het totaal aantal meter stof dat benodigd is voor deze intekening.

In het geval van het voorbeeld:

Intekening	Kleur	Cup	38	40	42	44	46	Lagen	Meters
1	601	B	2	2	2	0	0	13	2,71
2	601	B	0	3	1	4	2	2	4,9
3	601	C	2	4	4	2	0	10	5,7
4	601	C	0	0	0	4	4	2	4,25

De door de nester gemaakte intekeningen worden uitgeprint op de *plotter*.

***Plotter:***

Een plotter is een uitvoerapparaat waarmee (lijn)tekeningen kunnen worden gemaakt. Een plotter is dus een grote printer, waarmee hele intekeningen kunnen worden uitgeprint.

Deze prints worden samen met de snijlijst opgestuurd naar het buitenland, waar de stof daadwerkelijk gesneden wordt.

Daar worden per intekening eerst de lagen stof opgespannen, dit gebeurt met de hand. Vervolgens wordt de print van de intekening boven op de opgespannen lagen stof gelegd en daarop gestreken. Dan worden de opgespannen lagen stof met daarop de print van de intekening naar de snijtafel gebracht. Hier worden de patroondelen uitgesneden.

Vervolgens worden de uitgesneden patroondelen bij elkaar gezocht in elkaar gezet tot gehele items. De items die te veel worden uitgesneden (overproductie), worden ook daadwerkelijk in elkaar gezet en er wordt geprobeerd deze items nog te verkopen.

Momenteel worden de snijlijsten gemaakt door het programma e-fashion. Daarin moeten een paar waarden worden ingevuld.

- het aantal lagen stof dat maximaal tegelijkertijd gesneden kan worden. Dit aantal hangt af van de soort stof en de dikte en is meestal ongeveer 30.
- het aantal items dat maximaal in een intekening geplaatst mag worden. Dit aantal moet worden ingevuld om er voor te zorgen dat de intekening in zijn geheel op de snijtafel past. Voor een bikini, waar relatief weinig stof voor nodig is, is dit aantal groter dan voor een badpak, waar meer stof voor nodig is. Dit aantal wordt geschat uit ervaring.

Vervolgens kan dit computerprogramma een snijlijst genereren. Daarvoor gebruikt het de ingevulde waarden. Er worden voor elke kleur en cup-maat apart één of meerdere intekeningen gegenereerd. Beach Life is niet tevreden met de snijlijsten die door dit programma gegenereerd worden. Daarom worden ze meestal met de hand nog aangepast / verbeterd.

***Punten voor verbetering***

Op het eerste gezicht zijn er twee mogelijkheden tot verbetering. De eerste is het verbeteren van de planning van het snijden, hiermee kan het aantal intekeningen verminderd worden en daarmee de productietijd. De tweede is het efficiënter gebruiken van de stof.

**Verbeteren van de planning van het snijden**

Bij het maken van de snijlijst worden nu alle cupmaten, kleuren en werkorders apart genomen. Dit wil zeggen dat er eigenlijk voor elke werkorder, voor elke kleur, voor elke cupmaat een nieuwe snijlijst gemaakt wordt. Hierdoor is zijn er voor elke cup-kleur-product-combinatie één of meerdere intekeningen nodig. Vooral wanneer het om kleine aantallen gaat is dit systeem erg inefficiënt.

Cupmaten en werkorders kunnen relatief eenvoudig worden samengevoegd. Dit kan door elke andere cup-maat of product als een nieuwe maat te zien. Hierdoor kan het totaal aantal intekeningen beperkt worden.

Ook zouden kleuren kunnen worden samengevoegd; als eenzelfde product in meerdere kleuren geproduceerd moet worden, zouden lagen stof van verschillende kleuren op elkaar gelegd kunnen worden en tegelijkertijd gesneden.

### **Stofefficiëntie**

Bij het maken van de snijlijst wordt nu geen rekening gehouden met of de patroondelen goed in elkaar passen en dus of de stof wel efficiënt gebruikt wordt; eerst wordt de snijlijst gemaakt en vervolgens worden pas de intekeningen gemaakt met de nester. Ook worden nu altijd alle onderdelen van één badpak in dezelfde intekening gezet, terwijl het gunstiger zou kunnen zijn om verschillende patroondelen van een badpak in verschillende intekeningen te zetten, als de patroondelen dan beter in elkaar passen.

Om dit te verbeteren zou het maken van intekeningen met de nester geïntegreerd moeten worden in het maken van de snijlijst. De nester heeft echter nogal wat rekentijd nodig voor het maken van een intekening, dus deze optie is waarschijnlijk te tijdrovend.

Ook zou vooraf een grondig onderzoek met de nester gedaan kunnen worden om vast te stellen welke patroondelen meestal goed in elkaar passen en welke patroondelen slecht.

### **Criterionfunctie**

De kwaliteit van de snijplanning is een afweging tussen de productietijd, de hoeveelheid verbruikte stof en de overproductie. Bepalend voor de productietijd zijn het aantal intekeningen en het totaal aantal patroondelen dat uitgesneden moet worden. Al deze factoren moeten geminimaliseerd worden.

### **De productietijd**

Het snijden gebeurt in twee stappen: het opspannen van de stof en het daadwerkelijke snijden. Deze handelingen gebeuren op verschillende tafels, dus er kan al begonnen worden met het opspannen van de stof voor de volgende intekening, als de vorige intekening naar de snijtafel is. De tijd die nodig is voor het opspannen van de stof, hangt af van het aantal lagen stof dat opgespannen moet worden. De tijd die nodig is voor het daadwerkelijke snijden hangt af van het aantal patroondelen dat uitgesneden moet worden (de lengte van de intekening). Van deze 2 stappen kost de eerste, het opspannen van de lagen stof, de meeste tijd.

Omdat het opspannen van de lagen stof het meeste tijd kost, is het voor het snel verlopen van het snijproces van belang dat er met zo min mogelijk intekeningen gewerkt wordt. Daardoor hoeft zo min mogelijk het proces van opspannen van lagen stof doorlopen te worden. Daarnaast moeten de intekeningen zo min mogelijk items bevatten, zodat het snijden ook zo min mogelijk tijd kost.

### **De hoeveelheid verbruikte stof**

De hoeveelheid te gebruiken stof kan niet direct geoptimaliseerd worden, want hiervoor zouden alle mogelijke intekeningen getest moeten worden met de nester. Dit is te tijdrovend, omdat de nester enige rekentijd nodig heeft.

Aangezien de nester niet wordt gebruikt tijdens het maken van snijlijsten, is het niet gunstig om patroondelen van een badpak over meerdere intekeningen te verspreiden. Het eventuele stofvoordeel dat behaald zou kunnen worden, doordat patroondelen beter in elkaar passen,

kan nooit tijdens het optimaliseren bepaald worden, en kan ook nadelig uitvallen. Er valt hier dus geen voordeel te halen.

Bovendien is het in de praktijk waarschijnlijk ook onhandig om patroondelen van eenzelfde item over meerder intekeningen te verspreiden, omdat na het uitsnijden de patroondelen van een item op de juiste manier bij elkaar gezocht moeten worden, om ze vervolgens aan elkaar te naaien. Als alle onderdelen van eenzelfde item verspreid zitten over verschillende intekeningen kan het bij elkaar zoeken van de juiste patroondelen een lastige taak worden.

Stofverlies ontstaat door niet goed in elkaar passende patroondelen, of te weinig patroondelen in een intekening, zodat niet de gehele breedte van de stof benut wordt. Bovendien kost elke intekening 4 cm stof per laag, omdat aan beide kanten van de intekening 2 cm stof moet worden vrijgehouden.

Het stofverlies door niet passende patroondelen en te weinig patroondelen per intekening kan niet gemeten worden, want de hoeveelheid verbruikte stof wordt niet tijdens het optimaliseren van de snijlijst bepaald. Om toch enig inzicht in het stofverlies te krijgen is er een onderzoek gedaan naar het passen van patroondelen met de nester. Zie hiervoor bijlage A.

De conclusies van dit onderzoek zijn:

- Een minimum van vijf items per intekening zorgt ervoor dat de stofbreedte efficiënt gebruikt wordt.
- Wanneer verschillende producten en maten in een intekeningen worden gemixt, kost een product gemiddeld meestal minder stof dan wanneer intekeningen maar één product bevatten.

Voor het beperken van stofverbruik is het dus van belang dat de intekeningen minimaal vijf producten bevatten, zodat de stofbreedte goed gebruikt wordt. Het is zelfs gunstig om de intekeningen zo lang mogelijk te maken, omdat bij elke intekening per laag 4 cm stofverlies optreedt. Ook is het voordelig om verschillende maten en producten te mixen.

## **Overproductie**

Er wordt nog geprobeerd de te veel geproduceerde items te verkopen, maar dit is soms lastig, als het geen mooi weer is in de zomer. Beach Life vindt een overproductie van 3%, verspreid over de verschillende maten, acceptabel.

## **Samenvatting criteriumfunctie**

Voor de snelheid van het snijproces is het van belang om het aantal intekeningen te minimaliseren. Daarmee wordt bereikt dat een minimaal aantal malen het tijdrovende proces van lagen stof opgespannen moet worden doorlopen. Maar daarmee wordt ook bereikt dat er meer items per intekening komen en dat is gunstig voor het stofverbruik, want elke intekening kost 4 cm extra stof per laag.

Daarnaast moeten de intekeningen zo min mogelijk items bevatten, zodat het snijden zo min mogelijk tijd kost. Dit is in strijd het vorige criterium.

Dus bij het maken van een snijlijst moet gezocht worden naar een snijlijst met zo min mogelijk intekeningen. Hierbij moet er wel gezorgd worden dat er niet te veel overproductie ontstaat en dat de intekeningen te veel items bevatten.

## **Wiskundig model**

Er moet een snijlijst worden gegenereerd met zo min mogelijk intekeningen en tegelijkertijd niet te veel overproductie. Er kan dus gezocht worden naar een snijlijst met een zo laag mogelijke waarde voor de functie:

aantal intekeningen +  $c \cdot$  (aantal overproductie).

De waarde  $c$  is een gewicht dat aangeeft hoe belangrijk het beperken van overproductie is ten opzichte van het beperken van het aantal intekeningen.

Als bijvoorbeeld van te voren bekend is dat de te veel geproduceerde producten (overproductie) relatief makkelijk nog te verkopen zijn, dan kan voor  $c$  een lage waarde gekozen worden; is bekend dat er niks meer met te veel geproduceerde producten gedaan kan worden, dan wordt er voor  $c$  een hoge waarde gekozen.

## IP model

Dit probleem is te schrijven als een *IP* model.

### **IP:**

IP is een afkorting voor Integer Program. Een Integer Program is de geheeltallige versie van een LP (Linear Program). LP is een van de meest in de praktijk toegepaste wiskundige methoden. Bij LP wordt de optimale waarde bepaald van een lineaire criteriumfunctie van verschillende beslissingsvariabelen. Deze variabelen moeten voldoen aan een aantal lineaire bijvoorwaarden. LP problemen kunnen relatief eenvoudig worden opgelost met de simplex methode.

Bij IP moeten de variabelen geheeltallige waarden hebben. IP problemen zijn in tegenstelling tot LP problemen niet eenvoudig op te lossen.

De keuze voor een IP model is gemaakt, omdat het hier gaat om aantallen producten en aantallen intekeningen. Deze aantallen moeten altijd gehele getallen zijn, want halve producten en halve intekeningen zijn niet mogelijk. Daarom is het noodzakelijk dat een IP model wordt opgesteld en geen LP model.

Eerst wat definities voordat het model opgesteld kan worden:

### **De verzameling van typen (kleuren) stof:**

$i \in \{1, 2, \dots, I\}$ ,  $I$  = het aantal typen (kleuren) stof

### **De verzameling van items:**

$j \in \{1, 2, \dots, J\}$ ,  $J$  = het aantal verschillende items dat geproduceerd moet worden.

### **De verzameling van mogelijke intekeningen:**

$k \in \{1, 2, \dots, K\}$ ,  $K$  = het aantal mogelijke intekeningen.

$K$  is dus heel groot, omdat er heel veel verschillende combinaties van items mogelijk zijn die samen een intekening kunnen vormen.

### **De constanten (input):**

$c$  = constante voor in de criteriumfunctie, die aangeeft hoe kostbaar overproductie is,

$m_i$  = maximaal aantal lagen dat van stof type  $i$  tegelijk gesneden kan worden,

$d_{ij}$  = het aantal dat van item  $j$  van stof type (kleur)  $i$  geproduceerd moet worden,

$A_{jk}$  = het aantal keer dat item  $j$  voorkomt in mogelijke intekening  $k$ .

### **De variabelen:**

$x_{ik}$  = aantal lagen dat gesneden moet worden van intekening  $k$  met stof  $i$ ,

$y_{ij}$  = overproductie van item  $j$  van stof  $i$ ,

$z_{ij}$  = productie van item  $j$  van stof  $i$ ,

$g_k$  = aantal keer dat intekening  $k$  gebruikt wordt, dus  $g_k = 0$  als intekening  $k$  niet gebruikt wordt.



**Model:**

Minimaliseer:

$$\sum_{k=1}^K g_k + c \sum_{i=1}^I \sum_{j=1}^J y_{ij}.$$

Onder:

$$z_{ij} = d_{ij} + y_{ij} \quad \text{voor alle } i, j;$$

$$z_{ij} = \sum_{k=1}^K x_{ik} * A_{jk} \quad \text{voor alle } i, j;$$

$$g_k \geq \sum_{i=1}^I \frac{x_{ik}}{m_i} \quad \text{voor alle } k;$$

$$x_{ik}, y_{ij}, z_{ij}, g_k \in \mathbb{Z}^+ \quad \text{voor alle } i, j, k.$$

**Uitleg model:**

De criteriumfunctie is: aantal intekeningen +  $c$ \*(aantal overproductie).

Het aantal intekening is de som van het aantal keer dat elke intekening gebruikt wordt. Voor het totaal aantal overproductie worden van alle items en kleuren stoffen de aantallen overproductie opgeteld.

De eerste restrictie zorgt ervoor dat het aantal dat geproduceerd wordt van item  $i$  en kleur  $j$  hetzelfde is als het aantal dat daarvan geproduceerd moet worden + het aantal dat teveel geproduceerd wordt.

De tweede restrictie berekent de productie van item  $i$  van kleur  $j$ . Hiervoor wordt, van alle mogelijke intekeningen  $k$ , het aantal lagen dat wordt gesneden van intekening  $k$  met stof  $i$  vermenigvuldigd met het aantal keer dat item  $j$  voorkomt in intekening  $k$ . Deze waarden worden voor alle mogelijke intekeningen bij elkaar opgeteld.

De derde restrictie berekent het aantal keer dat intekening  $k$  gebruikt wordt,  $g_k$ . Er wordt alleen een ondergrens voor  $g_k$  gegeven, dit kan omdat  $g_k$  in de criteriumfunctie geminimaliseerd wordt. Daardoor zal  $g_k$  automatisch de kleinste toegestane waarde aannemen. De ondergrens voor  $g_k$  is het aantal lagen dat van intekening  $k$  gesneden moet worden, gedeeld door het maximaal aantal lagen waar tegelijkertijd doorheen gesneden mag worden. Dit aantal wordt voor alle kleuren stoffen  $i$  opgeteld. Deze formule werkt omdat  $g_k$  geheeltallig moet zijn. Zodra  $x_{ik}$ , voor een  $i$  groter dan 0 is, dan moet  $g_k$  minstens 1 zijn.  $g_k$  kan alleen 0 zijn, als voor alle  $i$   $x_{ik}$  ook 0 is, dus als intekening  $k$  helemaal niet gebruikt wordt.

Als laatste wordt geëist dat alle variabelen niet negatieve gehele getallen zijn. Alle variabelen zijn aantallen, dus daarom moeten ze geheeltallig en niet negatief zijn.

**Complexiteit**

De complexiteitstheorie is het gebied van wetenschappelijk onderzoek dat zich bezighoudt met de vragen welke wiskundige problemen al dan niet oplosbaar zijn en hoe efficiënt oplossingen voor een gegeven probleem zijn. De complexiteit van een gegeven algoritme wordt uitgedrukt in termen van de grootte van de invoer voor dat algoritme.

De makkelijke problemen zijn die problemen waarvoor een algoritme bekend is met een complexiteit die een *polynomiale* uitdrukking is in de grootte van de invoer. Deze problemen vormen samen de verzameling  $P$  van *polynomiaal* oplosbare problemen.

**Polynoom:**

Een polynoom is een uitdrukking waarin optelling en vermenigvuldiging een *eindig* aantal keren voorkomen, of die op die manier herschreven kan worden.

De moeilijke problemen zijn de problemen waarvoor een dergelijk algoritme niet bekend is. Een bepaalde klasse van moeilijke problemen waarvoor de complexiteitsgraad nog niet door een polynoom kan worden beschreven vormen de groep van NP-moeilijke problemen. Dit zijn problemen waarvoor geen polynomiaal algoritme bestaat, maar waarvan bij een eventuele oplossing wel in polynomiale tijd gecontroleerd kan worden of deze correct is (NP staat voor 'non-deterministische polynomiaal').

Een speciale groep hierbinnen zijn de NP-complete problemen. Van deze problemen is ook bewezen dat ieder probleem in de verzameling van NP-problemen omgeschreven kan worden naar een NP-compleet probleem. Dat wil zeggen, als je een algoritme had om een NP-compleet probleem op te lossen, zou je dat algoritme op de een of andere manier kunnen gebruiken om ieder probleem in NP op te lossen. Het betekent ook dat als er ooit een polynomiaal algoritme gevonden wordt dat een NP-compleet probleem oplost, dat alle NP-problemen in polynomiale tijd opgelost kunnen worden.

Wel zijn voor een groot aantal van deze 'computationeel moeilijke' problemen algoritmen bekend die niet met zekerheid de beste, maar wel een goede oplossing geven, zogenaamde *heuristische methoden*.

Het algemene IP probleem valt onder de klasse NP-moeilijk. Dat wil zeggen dat er geen algoritme bekend is dat elk IP probleem binnen polynomiale tijd op kan lossen. Dat wil niet zeggen dat geen enkel IP probleem binnen polynomiale tijd oplosbaar is; er zijn speciale gevallen van IP problemen waar wel een polynomiaal algoritme voor bekend is.

**Het oplossen van dit IP model:**

De input van dit model bestaat onder andere uit de matrix  $A$ , met daarin alle mogelijke intekeningen. Deze matrix is niet bekend en zal eerst gemaakt moeten worden.

Het aantal mogelijke intekeningen is heel groot, omdat er heel veel verschillende combinaties van items mogelijk zijn die samen een intekening kunnen vormen. Het aantal mogelijk intekeningen is veel te groot om de matrix  $A$  helemaal te vullen met alle mogelijke intekeningen.

Om dit probleem te omzeilen zou kolomgeneratie gebruikt kunnen worden. Daarbij worden eerst een groot aantal mogelijke intekeningen gegenereerd, die de matrix  $A$  zullen vormen en vervolgens wordt het IP model opgelost met alleen deze intekeningen. Dit IP model is dan dus een stuk kleiner dan het oorspronkelijke IP model. Doordat niet alle mogelijke intekeningen meer kunnen voorkomen is er dan geen garantie meer dat de oplossing die gevonden wordt optimaal is. Ook is niet duidelijk welke intekeningen gegenereerd moeten worden, zodat er binnen deze intekeningen een goede oplossing gevonden kan worden.

Vervolgens is het resulterende IP probleem nog steeds een groot IP probleem, waarvan niet direct duidelijk is hoe dat opgelost moet worden.

Voor het verzinnen van goede intekeningen en het oplossen van het IP probleem dat dan ontstaat zouden dus goede heuristieken verzonden moeten worden. Dit is waarschijnlijk lastiger dan voor en hele probleem in een keer een heuristiek verzinnen. Dus dit IP model ga ik niet gebruiken om het probleem op te lossen.

## Hoofdstuk 3 Heuristieken

Omdat het wiskundige model niet binnen redelijke tijd optimaal op te lossen is, heb ik er dus voor gekozen om dit probleem met een heuristiek op te lossen. Dat wil zeggen dat niet gegarandeerd de optimale oplossing gevonden wordt. Het voordeel hiervan is, dat in redelijk snelle tijd goede oplossingen gevonden kunnen worden.

In totaal zijn 6 heuristieken ontwikkeld, ieder met een andere functie. De eerste 3 zijn voor één kleur, de andere 3 voor meerder kleuren.

De eerste heuristiek is een constructieheuristiek; deze wordt gebruikt om een startoplossing te maken. Vervolgens wordt een verbeteringsheuristiek ontwikkeld; deze probeert de startoplossing te verbeteren, en creëert een aantal nieuwe oplossingen, waarvan sommigen beter en sommigen slechter zijn dan de startoplossing. De derde heuristiek gaat over overproductie. De voorgaande heuristieken zijn zo gemaakt dat ze precies aan de behoefte voldoen. Maar soms is het gunstig om wat overproductie te hebben, als dit beter uitkomt met snijden. Deze overproductie heuristiek is een verbeteringsheuristiek die probeert een oplossing te verbeteren door enige overproductie toe te staan.

Vaak worden dezelfde producten in meerdere kleuren geproduceerd. Deze verschillende kleuren stof kunnen dus tegelijkertijd gesneden worden in 1 intekening, door lagen van verschillende kleuren stof op elkaar te leggen. Voor deze situatie zijn ook een constructieheuristiek en twee verbeteringsheuristieken ontwikkeld.

In de derde paragraaf wordt uitgelegd hoe uit alle oplossingen die door de verschillende heuristieken gecreëerd worden, de beste gekozen kan worden.

Als laatste wordt in dit hoofdstuk beschreven wat de zwakke punten van de heuristieken zijn.

### Eén kleur

#### Constructieheuristiek

De constructieheuristiek is erop gebaseerd om zo snel mogelijk zo veel mogelijk items uit te snijden. Bij elke iteratie wordt een nieuwe intekening gemaakt. Eerst wordt gekeken bij welk aantal lagen de meeste items uitgesneden kunnen worden. Dit aantal lagen wordt gekozen en daarmee wordt de intekening bepaald. De items die daarmee gesneden kunnen worden, worden van de behoefte afgetrokken.

#### Input

*behoefte(i)* is het aantal dat van item *i* geproduceerd moet worden.

*maxlagen* is het maximaal aantal lagen dat tegelijkertijd gesneden kan worden.

*maxitems* is het maximaal aantal items dat in een intekening geplaatst mag worden.

#### Stappenplan

Stap 0:  $intekeningen = 0$ .

Stap 1:  $intekeningen = intekeningen + 1$ .

$beste = 0$ .

$lagen = maxlagen$ .

Stap 2: Bepaal hoeveel items er in de intekening passen indien *lagen* wordt gekozen als aantal lagen:

$$aantal = \sum_{items\ i} \text{floor}(behoefte(i)/lagen).$$

Als  $aantal > maxitems$  dan  $aantal = maxitems$ .

Het aantal producten dat met deze intekening geproduceerd kan worden is dan:

$$totaal = aantal * lagen.$$

Stap 3: Als  $totaal > beste$  dan  $beste = totaal$ ,  $bestelagen = lagen$ .

$lagen = lagen - 1$ .

Als  $lagen > 0$  ga naar stap 2, anders ga naar stap 4.

Stap 4:  $totaalaantal = 0$ .  
 Voor alle items  $i$ :  
 $aantal(i) = \min(maxitems - totaalaantal, \text{floor}(behoefte(i)/bestelagen))$ .  
 $totaalaantal = totaalaantal + aantal(i)$ .  
 $behoefte(i) = behoefte(i) - bestelagen * aantal(i)$ .

De intekening met nummer  $intekeningen$  is nu bekend: van item  $i$  staan er  $aantal(i)$  in de intekening en van deze intekening worden  $bestelagen$  lagen gesneden.

Als voor alle items  $i$ :  $behoefte(i) = 0$ : STOP (er is een oplossing gevonden), anders: ga naar stap 1.

## Uitleg

Stap 0: er wordt begonnen met 0 intekeningen. Dan wordt telkens met stappen 1 tot en met 4 een intekening bijgemaakt. Als aan het eind van stap 4 aan de volledige behoefte van de werkorder voldaan is, dan stopt de heuristiek. Als nog niet volledig aan de behoefte voldaan is, wordt weer teruggegaan naar stap 1 en wordt een nieuwe intekening gemaakt.

De stappen 1 tot en met 4 creëren dus telkens een nieuwe intekening. In stap 1 wordt het aantal intekeningen ( $intekeningen$ ) met 1 verhoogd. Ook worden de variabelen  $beste$  en  $lagen$  geïnitieerd. Deze variabelen worden gebruikt in de stappen 2 en 3. In deze stappen wordt bepaald hoeveel lagen van deze nieuwe intekening gesneden moeten worden. In stap 4 wordt vervolgens bepaald welke items hoe vaak in de intekening voorkomen.

Stappen 2 en 3 vormen de kern van het algoritme. Bij deze stappen wordt het aantal lagen van de intekening bepaald. Dit moet het aantal lagen zijn, waarbij met deze intekening in totaal zoveel mogelijk items uitgesneden worden. Daartoe worden alle mogelijke aantallen lagen uitgetest, dat zijn alle aantallen tussen  $maxlagen$  en 1. Voor elk aantal lagen wordt in stap 2 bepaald hoeveel items er met dit aantal lagen in deze intekening uitgesneden kunnen worden. Om dit te bepalen wordt voor elk item bepaald hoeveel stuks van dit item maximaal in de intekening kunnen, zonder dat er teveel geproduceerd wordt. Dit kan berekend worden met  $\text{floor}(behoefte(i)/lagen)$ , met  $\text{floor}(x)$  de functie die het grootste gehele getal kleiner of gelijk aan  $x$  geeft. Deze aantallen van alle items bij elkaar opgeteld geeft het aantal items dat in de intekening kan. Deze mag niet groter zijn dan  $maxitems$ . Dus als dit aantal groter is, dan wordt het gelijk gesteld aan  $maxitems$ . Om vervolgens het totaal aantal items dat gesneden wordt te berekenen, wordt het aantal items in de intekening vermenigvuldigd met het aantal lagen. In stap 3 wordt daarna gekeken of dit totaal aantal ( $totaal$ ) beter is dan het tot dan toe beste totaal aantal ( $beste$ ). Als dit zo is dan  $bestelagen = lagen$  en  $beste = totaal$ . Vervolgens wordt weer naar het volgende aantal lagen keken, als dit nog groter dan 0 is ga je weer naar stap 2, anders ga je naar stap 4 en is het aantal lagen bepaald. Er wordt van  $maxlagen$  naar beneden gezocht, zodat wanneer er 2 lagenaantallen hetzelfde totaal aantal items geven, het hoogste lagenaantal gekozen wordt.

In stap 4 wordt de intekening bepaald, daartoe wordt eerst het totaal aantal items in de intekening komt ( $totaalaantal$ ) geïnitieerd. Vervolgens worden alle items  $i$  langsgegaan. Voor elk item  $i$  wordt eerst berekend hoe vaak dit item in deze intekening voorkomt ( $aantal(i)$ ). Dit aantal is het minimum van het aantal plaatsen dat nog vrij is in de intekening ( $maxitems - totaalaantal$ ) en het maximaal aantal dat van dit item in deze intekening kan komen zonder dat te veel geproduceerd wordt ( $\text{floor}(behoefte(i)/bestelagen)$ ). Vervolgens wordt dit aantal opgeteld bij het totaal. Uiteindelijk wordt het aantal dat geproduceerd wordt van item  $i$  ( $bestelagen * aantal(i)$ ) afgetrokken van de behoefte.

Het laatste deel van stap 4 bepaalt of de heuristiek gestopt kan worden of dat weer begonnen wordt met stap 1.

## Verbeteringsheuristiek 1: ander lagenaantal

De constructieheuristiek geeft een goede startoplossing. Om deze te verbeteren moeten andere aantallen lagen uitgeteerd worden. Om het snijproces te versnellen is het vaak goed om meerdere lagen tegelijk te snijden, zodat er minder items in de intekening zitten. Ook is het mogelijk dat het beter uitkomt als soms minder lagen gesneden worden.

Deze verbeteringsheuristiek probeert betere oplossingen te genereren door niet steeds het aantal lagen te kiezen waarmee de meeste items in een keer geproduceerd worden, maar soms een ander aantal lagen.

Als er een startoplossing is, kan het aantal lagen bij de eerste intekening veranderd worden, maar dit kan ook bij de tweede intekening gebeuren, of bij de derde. Op één startoplossing kan de verbeteringsheuristiek dus meerdere keren toegepast worden.

Een door de constructieheuristiek geconstrueerde snijlijst zit zo ingewikkeld in elkaar dat het niet mogelijk is om zomaar bij een intekening het aantal lagen te veranderen. Een kleine wijziging ergens in de snijlijst, heeft grote gevolgen voor de productie. Om er voor te zorgen dat het aantal items dat geproduceerd wordt hetzelfde blijft als de behoefte, moeten alle andere intekening ook gewijzigd worden. Daarom is het beter om alle intekeningen na de gewijzigde intekening helemaal opnieuw te bepalen. Daarvoor wordt weer de constructieheuristiek gebruikt.

De verbeteringsheuristiek is dus een soort uitbreiding op de constructieheuristiek. Er wordt, net zoals in de constructieheuristiek, telkens een nieuwe intekening gemaakt en dan worden de geproduceerde items van de behoefte afgetrokken. De input voor deze verbeteringsheuristiek is hetzelfde als de input voor de constructieheuristiek, plus een *startoplossing* en het *intekeningnummer* van de intekening waarbij een ander aantal lagen gekozen moet worden. Bij het verbeteren van de *startoplossing* worden de eerste intekeningen, tot de intekening met nummer *intekeningnummer*, hetzelfde genomen als de eerste intekeningen van de *startoplossing*. Bij de intekening met nummer *intekeningnummer*, wordt het aantal lagen gekozen waarmee de meeste items geproduceerd worden met de restrictie dat dit een ander aantal lagen is dan het aantal lagen bij die intekening van de *startoplossing*. Na deze intekening worden de rest van de intekeningen helemaal opnieuw bepaald volgens de constructieheuristiek.

### Input

*behoefte(i)* is het aantal dat van item *i* geproduceerd moet worden.

*maxlagen* is het maximaal aantal lagen dat tegelijkertijd gesneden kan worden.

*maxitems* is het maximaal aantal items dat in een intekening geplaatst mag worden.

*startoplossing* is de oplossing die verbeterd moet worden.

*intekeningnummer* is het nummer van de intekening waarbij een ander aantal lagen gekozen moet worden.

### Stappenplan

Stap 0:  $intekeningen = 0$ .

$startlaag(j)$  = het aantal lagen dan van intekening *j* gesneden wordt in *startoplossing*.

Stap 1:  $intekeningen = intekeningen + 1$ .

Als  $intekeningen < intekeningnummer$ :

$bestelagen = startlaag(intekeningen)$ . Ga naar stap 4.

Anders:

$beste = 0$ ,  $lagen = maxlagen$ . Ga naar stap 2.

- Stap 2: Bepaal hoeveel items er in de intekening passen indien *lagen* wordt gekozen als aantal lagen:  

$$aantal = \sum_{items\ i} \text{floor}(\text{behoefte}(i)/\text{lagen}).$$
 Als  $aantal > \text{maxitems}$  dan  $aantal = \text{maxitems}$ .  
 Het aantal producten dat met deze intekening geproduceerd kan worden is dan:  

$$\text{totaal} = \text{aantal} * \text{lagen}.$$
- Stap 3: Als  $\text{intekeningen} = \text{intekeningnummer}$  en  
 $\text{lagen} = \text{startlaag}(\text{intekeningnummer})$ :  $\text{totaal} = 0$ .  
 Als  $\text{totaal} > \text{beste}$  dan  $\text{beste} = \text{totaal}$ ,  $\text{bestelagen} = \text{lagen}$ .  
 $\text{lagen} = \text{lagen} - 1$ .  
 Als  $\text{lagen} > 0$ : ga naar stap 2, anders: ga naar stap 4.
- Stap 4:  $\text{totaalaantal} = 0$ .  
 Voor alle items  $i$ :  

$$aantal(i) = \min(\text{maxitems} - \text{totaalaantal}, \text{floor}(\text{behoefte}(i)/\text{bestelagen})).$$

$$\text{totaalaantal} = \text{totaalaantal} + \text{aantal}(i).$$

$$\text{behoefte}(i) = \text{behoefte}(i) - \text{bestelagen} * \text{aantal}(i).$$
 De intekening met nummer  $\text{intekeningen}$  is nu bekend: van item  $i$  staan er  $\text{aantal}(i)$  in de intekening en van deze intekening worden  $\text{bestelagen}$  lagen gesneden.  
 Als voor alle items  $i$ :  $\text{behoefte}(i) = 0$ : STOP (er is een oplossing gevonden), anders: ga naar stap 1.

### Uitleg

Omdat de verbeteringsheuristiek een uitbreiding is van de constructieheuristiek, lijkt dit stappenplan ook veel op het stappenplan van de constructieheuristiek. Wederom wordt steeds in stap 1 tot en met 4 een intekening gemaakt, en als aan de behoefte voldaan is wordt gestopt. Er zijn echter twee belangrijke verschillen tussen de verbeteringsheuristiek en de constructieheuristiek.

1. De eerste intekeningen, tot aan de intekening die veranderd moet worden, moeten niet opnieuw bedacht worden, maar kunnen gekopieerd worden van de *startoplossing*.
2. De intekening die veranderd moet worden, moet een ander aantal lagen hebben dan de oorspronkelijke intekening.

Het eerste punt wordt verzorgd in stap 1. Als het aantal intekeningen (*intekeningen*) kleiner is dan *intekeningnummer*, dan moet de intekening gekopieerd worden van *startoplossing*. Het aantal lagen dat gekozen wordt voor deze intekening hoeft dus niet berekend te worden met stappen 2 en 3, maar kan gelijk genomen worden aan het aantal lagen van deze intekening in de *startoplossing* (*startlaag(intekeningen)*). Dus *bestelagen* wordt gelijkgesteld aan *startlaag(intekeningen)*. Vervolgens worden stappen 2 en 3 overgeslagen en wordt gelijk stap 4 gedaan.

Als het aantal intekeningen groter of gelijk is aan het *intekeningnummer*, dan wordt gewoon verder gegaan met stap 2.

Het tweede verschil komt tot uiting in stap 3. Bij de intekening die veranderd moet worden, worden wel de stappen 2 en 3 doorlopen om het nieuwe aantal lagen te bepalen. Echter, er moet een ander aantal lagen gekozen worden dan bij de *startoplossing*. Dus, als het aantal intekeningen (*intekeningen*) gelijk is aan *intekeningnummer* en het aantal lagen (*lagen*) gelijk is aan het te veranderen lagenaantal (*startlaag(intekeningnummer)*), dan wordt het totaal aantal items dat geproduceerd kan worden met dit lagenaantal (*totaal*) op 0 gezet, zodat dit aantal lagen niet gekozen kan worden. Zo wordt dus het op één na beste aantal lagen gekozen.

### Het toepassen van de verbeteringsheuristiek

Om een oplossing te verbeteren kan, zoals gezegd, bij elke intekening het aantal lagen veranderd worden, dus bij een oplossing kan voor elke intekening een nieuwe oplossing gemaakt worden.

Het is alleen niet zinvol om voor de laatste intekening in een oplossing deze verbeteringsheuristiek uit te voeren. Dit komt doordat het aantal lagen in de constructieheuristiek steeds zo gekozen wordt dat er zoveel mogelijk items geproduceerd worden en dat met zoveel mogelijk lagen. In de laatste intekening worden dus alle resterende items geproduceerd, met zoveel mogelijk lagen. Dit kan nooit worden verbeterd met een ander lagenaantal.

Op de startoplossing kan dus voor elke intekening, behalve de laatste, deze verbeteringsheuristiek worden toegepast. De nieuwe oplossingen kunnen beter of slechter zijn dan de oorspronkelijke oplossing. Op elke nieuwe oplossing zou vervolgens weer voor elke intekening, behalve de laatste, de verbeteringsheuristiek kunnen worden toegepast. Zo kan er stap voor stap gezocht worden naar steeds weer nieuwe oplossingen, die misschien beter zijn dan de eerder gevonden oplossingen. Maar als dit zo gedaan zou worden, dan zouden steeds dezelfde oplossingen gevonden worden. Immers, als eerst bij de tweede intekening een ander lagenaantal is gekozen en vervolgens bij deze nieuwe oplossing de eerste intekening veranderd zou worden, dan moet de tweede intekening alsnog opnieuw bepaald worden en was de eerste stap zinloos.

Daarom is er besloten om volgens het schema te zoeken dat op de bladzijde 25 staat. In dit schema staat elk rijtje met vierkantjes voor een gevonden snijlijst. Elk vierkantje is een intekening. Bij deze intekening staat het nummer van de intekening, het aantal lagen dat van deze intekening gesneden moet worden en het aantal lagen dat bij het maken van deze intekening niet gekozen mocht worden.

De bovenste snijlijst stelt de startoplossing voor. Deze heeft  $n$  intekeningen. Bij elke intekening staat een letter voor het aantal lagen dat van deze intekening gesneden moet worden. Bij "Niet:" staat steeds een streepje, omdat bij de startoplossing alle aantallen lagen vrij gekozen zijn.

Op de eerste  $n-1$  intekeningen wordt de verbeteringsheuristiek toegepast. Dit levert weer  $n-1$  nieuwe snijlijsten op. Bij de eerste van de nieuwe snijlijsten, snijlijst 2, is de verbeteringsheuristiek toegepast op de eerste intekening. Daarbij mag  $a$  niet als aantal lagen gekozen worden. De rest van de intekeningen is weer vrij gekozen volgens de constructieheuristiek. Bij de tweede van de nieuwe snijlijsten, snijlijst 3, is de verbeteringsheuristiek toegepast op de tweede intekening. De eerste intekening is overgenomen van de startoplossing en heeft  $a$  lagen, dit blokje is grijs gekeurd. De tweede intekening moet veranderd worden, daar mag dus niet  $b$  als aantal lagen gekozen worden. Vanaf de derde intekening worden de intekeningen weer vrij bepaald volgens de constructieheuristiek. Bij snijlijst 4 geldt hetzelfde verhaal: intekeningen 1 tot en met  $n-2$  zijn overgenomen van de startoplossing, deze blokjes zijn weer grijs gekleurd. Intekening  $n-1$  mag niet  $c$  lagen hebben, en vanaf intekening  $n$  wordt het aantal lagen weer vrij bepaald.

De volgende laag oplossingen komt weer volgens dezelfde methode tot stand. Alleen als eenzelfde intekeningnummer 2 keer veranderd wordt, dan zijn beide voorgaande lagenaantallen verboden. Dit is te zien in snijlijst 5. Daar wordt weer de verbeteringsheuristiek toegepast op de eerste intekening. Nu mag zowel  $a$  (van snijlijst 1) als  $e$  (van snijlijst 2) niet als aantal lagen gekozen worden bij de eerste intekening.

Ook is belangrijk om op te merken dat op intekeningen die zijn overgenomen van een vorige snijlijst (deze zijn grijs gekleurd in het schema) nooit de verbeteringsheuristiek wordt toegepast. Dit is te zien bij snijlijst 3 en snijlijst 4. Bij snijlijst 3 wordt de

verbeteringsheuristiek niet toegepast op de eerste intekening. Als dit wel zou gebeuren, dan zou precies dezelfde snijlijst ontstaan als snijlijst 2. Bij snijlijst 4 is ook te zien dat op de eerste  $n-2$  intekeningen niet de verbeteringsheuristiek wordt toegepast.

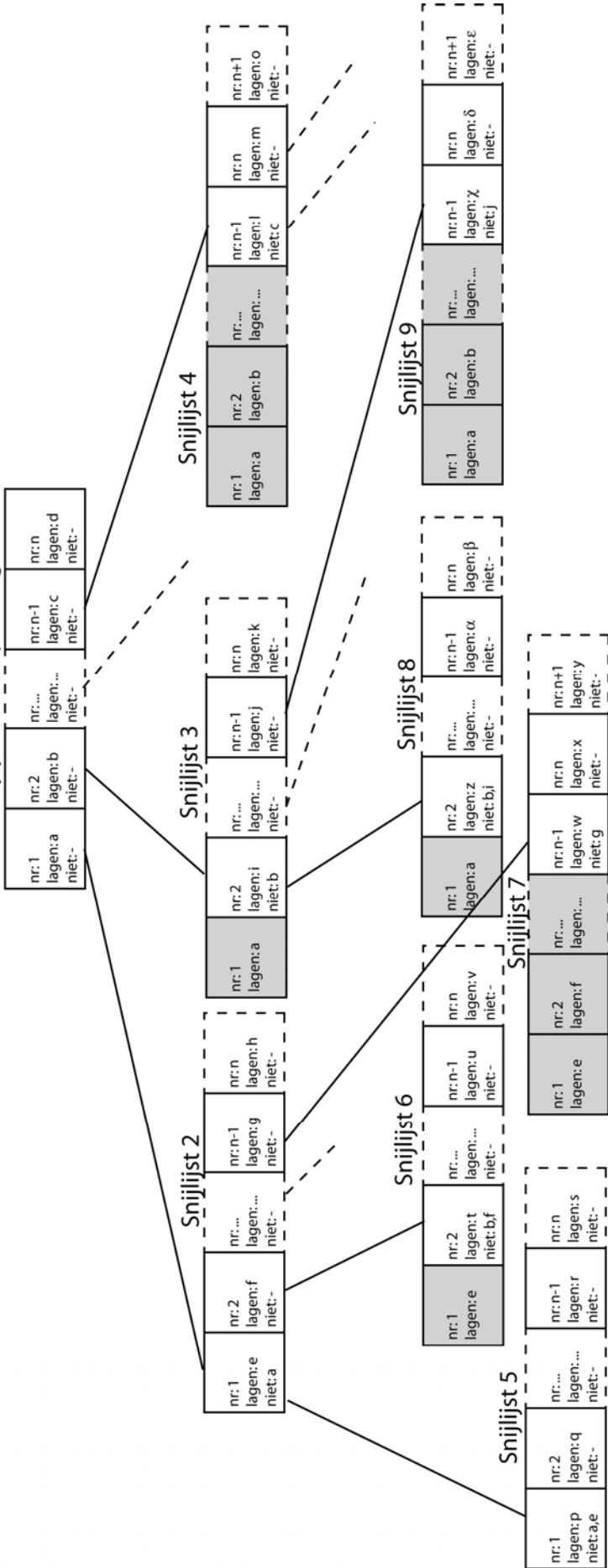
Dit proces van steeds een snijlijst veranderen kan lang doorgaan, zodat een hele boom met snijlijsten ontstaat. Ik heb er voor gekozen om steeds door te gaan met zoeken in een volgende laag, totdat de oplossing meer dan 3 keer verslechterd is ten opzichte van de vorige oplossing. Een oplossing wordt als slechter beschouwd als hij meer intekeningen bevat dan zijn voorganger, of als hij even veel intekeningen bevat, maar meer items moeten worden uitgesneden.

Dus als snijlijst 2 slechter is dan de startoplossing, en snijlijst 5 weer slechter is dan snijlijst 2 en de volgende 2 oplossingen ook weer slechter zijn dan hun voorganger, dan moet die tak van de boom gestopt worden. Als snijlijst 3 beter is dan de startoplossing en snijlijst 8 weer beter dan snijlijst 3, dan mag in die tak van de boom nog 3 keer de oplossing verslechterd worden, voordat gestopt wordt met zoeken.

Zo ontstaat een groep met snijlijsten, waaruit de gunstigste gekozen kan worden.



**Snijlijst 1: startoplossing**



## Verbeteringsheuristiek 2: overproductie

De bovenstaande heuristieken geven oplossingen die precies aan de behoefte voldoen. Soms kan het echter gunstig zijn om iets te veel te produceren, als dit voor het snijden veel gunstiger uitkomt.

Ook kijken de voorgaande heuristieken niet vooruit. Ze maken één voor één intekeningen die op dat moment het beste zijn, zonder te kijken wat dit betekent voor de behoefte die je daarna overhoudt. Hierdoor is er meestal een laatste intekening met maar 1 laag, die alle resterende behoefte bevat. Soms zitten er maar weinig items in deze laatste intekening en is het mogelijk om deze items al in de voorlaatste intekening, waar meer lagen van gesneden worden, te zetten, zodat de laatste intekening geschrapt kan worden. Wel wordt van deze items dan te veel geproduceerd, maar als dit niet om een te groot percentage gaat is dit acceptabel.

### Input

$behoefte(i)$  is het aantal dat van item  $i$  geproduceerd moet worden.

$maxitems$  is het maximaal aantal items dat in een intekening geplaatst mag worden.

$startoplossing$  is de oplossing die verbeterd moet worden.

### Stappenplan

Stap 0: Als  $startoplossing$  maar 1 intekening heeft: STOP.

Stap 1:  $totaalItems = \sum_{items\ i} behoefte(i)$ ,  
 $laatsteItem(i)$  = aantal item  $i$  in de laatste intekening van  $startoplossing$ ,  
 $laatsteItems = \sum_{items\ i} laatsteItem(i)$ .  
Als  $laatsteItems > (totaalItems/10)$ : STOP.

Stap 2:  $vorigeItem(i)$  = aantal item  $i$  in de voorlaatste intekening van  $startoplossing$ ,  
 $vorigeItems = \sum_{items\ i} vorigeItem(i)$ .  
 $vorigeItemsNieuw = vorigeItems$ .  
 $vorigeLagen$  = het aantal lagen van de voorlaatste intekening.  
 $laatsteLagen$  = het aantal lagen van de laatste intekening.  
Voor alle items  $i$ :

$$vorigeItemsNieuw = vorigeItemsNieuw + \text{ceil}(\text{laatsteItem}(i) * \text{laatsteLagen} / \text{vorigeLagen}).$$

Als  $vorigeItemsNieuw > maxitems$ : STOP.

Stap 3: Het is mogelijk om een nieuwe oplossing te maken met een intekening minder. Alle intekeningen tot de voorlaatste intekening zijn hetzelfde als in  $startoplossing$ . Van de voorlaatste en laatste intekening wordt 1 intekening gemaakt, met het aantal lagen van de voorlaatste intekening ( $vorigeLagen$ ). Voor elk item  $i$  is het aantal keer dat dit item voorkomt in deze intekening  $vorigeItem(i) + \text{ceil}(\text{laatsteItem}(i) * \text{laatsteLagen} / \text{vorigeLagen})$ .

### Uitleg

In deze heuristiek worden in de stappen 0 tot en met 2 controles gedaan of het mogelijk is om de laatste intekening te schrappen.

In stap 0 wordt gecontroleerd of er wel meerdere intekeningen zijn. Als er maar één intekening is, kan deze niet verwijderd worden en stopt het algoritme.

In stap 1 wordt gekeken hoeveel items in de laatste intekening zitten. Dit aantal is het minimum aantal overproductie dat zal ontstaan wanneer deze intekening bij een andere intekening gevoegd wordt. Als dit minimaal aantal overproductie meer dan 10% van het totaal aantal items is, dan stopt het algoritme, want meer dan 10% overproductie is te veel.

In stap 2 wordt gekeken of de items uit de laatste intekening wel bij de voorlaatste intekening passen. Om dit te bekijken wordt eerst bepaald hoeveel items de voorlaatste intekening al bevat ( $vorigeItems = \sum_{items\ i} vorigeItem(i)$ ).

Dan wordt voor elk item daarbij opgeteld, het aantal dat van dat item erbij moet, om de laatste intekening te vervangen. Dit wordt berekend met de formule  $ceil(laatsteItem(i) * laatsteLagen / vorigeLagen)$ , met  $ceil(x)$  de functie die het kleinste gehele getal groter of gelijk aan  $x$  geeft.

$laatsteItem(i) * laatsteLagen$  geeft het totaal aantal dat van item  $i$  in de laatste intekening geproduceerd wordt. Dit aantal moet in de voorlaatste intekening extra geproduceerd worden. Om te weten hoe vaak dit item in de intekening erbij moet, moet dit aantal gedeeld worden door het aantal lagen dat de voorlaatste intekening heeft ( $vorigeLagen$ ) en naar boven afgerond met de functie  $ceil$ .

$vorigeItemsNieuw$  is nu het aantal items dat in de intekening zit die de twee laatste intekeningen combineert. Als  $vorigeItemsNieuw$  groter is dan het maximaal aantal items dat in één intekening mag ( $maxitems$ ), dan is de nieuwe intekening dus te groot geworden. Dit is niet toegestaan en het algoritme stopt.

Als stap 3 bereikt wordt, dan is aan alle voorwaarden voldaan. Er kan een nieuwe oplossing gemaakt worden.

### **Het toepassen van de verbeteringsheuristiek**

Deze heuristiek kan worden toegepast op alle snijlijsten die met het uitvoeren van de constructieheuristiek en de eerste verbeteringsheuristiek zijn ontstaan.

### **Meerdere kleuren**

Vaak worden dezelfde producten in meerdere kleuren geproduceerd. Dat wil zeggen dat precies dezelfde patroondelen in verschillende kleuren stoffen geproduceerd moeten worden. In dit dat geval is het mogelijk om bij het opspannen van de stof, verschillende kleuren stof op elkaar te spannen. Deze verschillende kleuren stof kunnen tegelijkertijd gesneden worden met dezelfde intekening.

### **Constructieheuristiek**

Er is voor meerdere kleuren een constructieheuristiek ontwikkeld die hetzelfde principe toepast als de constructieheuristiek met één kleur.

Eerst wordt gekeken bij welk aantal lagen de meeste items uitgesneden kunnen worden. Echter, nu zijn er meerdere aantallen lagen, namelijk voor elke kleur kan een nieuw aantal lagen opgespannen worden. Er moet dus een lagencombinatie gekozen worden, waarmee de meeste items uitgesneden kunnen worden. Deze lagencombinatie wordt gekozen en daarmee wordt de intekening bepaald. De items die daarmee gesneden kunnen worden, worden van de behoefte afgetrokken.

### **Input**

$behoefte(i,k)$  is het aantal dat van item  $i$  en kleur  $k$  geproduceerd moet worden.

$maxlagen$  is het maximaal aantal lagen dat tegelijkertijd gesneden kan worden.

$maxitems$  is het maximaal aantal items dat in een intekening geplaatst mag worden.

### **Stappenplan**

Stap 0:  $intekeningen = 0$ .

Stap 1:  $intekeningen = intekeningen + 1$ .  
 $beste = 0$ .

Stap 2: Ga alle mogelijke lagencombinaties af.

$lagencombi(k)$  is het aantal lagen van kleur  $k$ .

Een geldige lagencombinatie moet voldoen aan:

$$\sum_{kleuren\ k} lagencombi(k) \leq maxlagen,$$

$$\sum_{kleuren\ k} lagencombi(k) > 0.$$

Voer stap 3 uit voor elke geldige lagencombinatie, ga daarna naar stap 4.

Stap 3: Bepaal hoeveel items er in de intekening passen met deze lagencombinatie.

Voor elk item  $i$  en elke kleur  $k$ :

Als  $lagencombi(k) > 0$ :

$$aantal(i,k) = \text{floor}(\text{behoefte}(i)/lagencombi(k)),$$

anders:

$$aantal(i,k) = maxitems.$$

Voor elk item  $i$ :

$$aantal(i) = \min_{kleuren\ k} aantal(i,k).$$

$$aantal = \sum_{items\ i} aantal(i).$$

Als  $aantal > maxitems$  dan  $aantal = maxitems$ .

Het aantal producten dat met deze intekening geproduceerd kan worden is dan:

$$totaal = aantal * \sum_{kleuren\ k} lagencombi(k).$$

Als  $totaal > beste$  dan  $beste = totaal$ ,  $bestelagencombi = lagencombi$ .

Stap 4:  $totaalaantal = 0$ .

Voor elk item  $i$  en elke kleur  $k$ :

Als  $bestelagencombi(k) > 0$ :

$$aantal(i,k) = \text{floor}(\text{behoefte}(i)/bestelagencombi(k)),$$

anders:

$$aantal(i,k) = maxitems.$$

Voor alle items  $i$ :

$$aantal(i) = \min(maxlagen - totaalaantal, \min_{kleuren\ k} aantal(i,k)).$$

$$totaalaantal = totaalaantal + aantal(i),$$

voor alle  $j$ :  $behoefte(i,k) = behoefte(i,k) - bestelagencombi(k) * aantal(i)$ .

De intekening met nummer  $intekeningen$  is nu bekend: van item  $i$  staan er  $aantal(i)$  in de intekening en van deze intekening worden  $bestelagencombi(k)$  lagen gesneden van kleur  $k$ .

Als voor alle items  $i$  en kleuren  $k$ :  $behoefte(i,k) = 0$ : STOP (er is een oplossing gevonden),

anders: ga naar stap 1.

## Uitleg

Omdat deze constructieheuristiek hetzelfde principe heeft als de constructieheuristiek met één kleur, lijkt dit stappenplan ook veel op het stappenplan van deze constructieheuristiek. Wederom wordt steeds in stap 1 tot en met 4 een intekening gemaakt, en als aan de behoefte voldaan is wordt gestopt. Alleen, hier wordt in de stappen 2 en 3 niet het aantal lagen bepaald, maar de lagencombinatie, waarbij de meeste items kunnen worden geproduceerd.

In stappen 2 en 3 worden alle mogelijke lagencombinaties uitgeprobeerd. Dat zijn alle combinaties, waarbij het totaal aantal lagen ( $\sum_{kleuren\ k} lagencombi(k)$ ) tussen  $maxlagen$  en 1 ligt. Voor elke geldige lagencombinatie wordt in stap 3 bepaald hoeveel producten er met deze combinatie in deze intekening uitgesneden kunnen worden.

Om dit te bepalen wordt voor elk item en elke kleur bepaald hoeveel stuks van dit item en die kleur maximaal in de intekening kunnen, zonder dat er teveel geproduceerd wordt. Dit kan berekend worden met  $\text{floor}(\text{behoefte}(i)/lagencombi(k))$ . Dit is dezelfde formule als bij de heuristiek voor één kleur, alleen nu moet het voor elke kleur  $k$  apart bepaald worden. Als

kleur  $k$  niet voorkomt in deze lagencombinatie ( $lagencombi(k) = 0$ ), dan vormt deze kleur dus geen beperking op het aantal keer dat dit item voor mag komen in de intekening, dus dan wordt het aantal gezet op  $maxitems$ . Voor elk item is het aantal stuks dat van dit item maximaal in de intekening kan, het minimum van deze aantallen over alle kleuren.

Deze aantallen van alle items bij elkaar opgeteld geeft het aantal items dat in de intekening kan. Deze mag niet groter zijn dan  $maxitems$ . Dus als dit aantal groter is, dan wordt het gelijk gesteld aan  $maxitems$ . Om vervolgens het totaal aantal producten dat gesneden wordt te berekenen, wordt het aantal items in de intekening vermenigvuldigd met het totaal aantal lagen ( $\sum_{kleuren k} lagencombi(k)$ ). Daarna wordt gekeken of dit *totaal* aantal beter is dan het tot dan toe beste totaal aantal (*beste*). Als dit zo is dan  $bestelagencombi = lagencombi$  en  $beste = totaal$ .

In stap 4 wordt de intekening bepaald, daartoe wordt eerst het totaal aantal items in de intekening komt (*totaalaantal*) geïnitieerd. Vervolgens worden alle items  $i$  langsgegaan. Voor elk item  $i$  wordt eerst berekend hoe vaak dit item in deze intekening voorkomt ( $aantal(i)$ ).

Dit aantal is het minimum van het aantal plaatsen dat nog vrij is in de intekening ( $maxitems - totaalaantal$ ) en het maximaal aantal dat van dit item in deze intekening kan komen zonder dat te veel geproduceerd wordt, zoals dit ook in stap 3 berekend wordt. Vervolgens wordt dit aantal opgeteld bij het totaal. Uiteindelijk wordt het aantal dat geproduceerd wordt van item  $i$  en kleur  $k$  ( $bestelagencombi(k) * aantal(i)$ ) afgetrokken van de behoefte.

Het laatste deel van stap 4 bepaalt of de heuristiek gestopt kan worden of dat weer begonnen wordt met stap 1.

Nadeel van dit algoritme is dat het alle lagencombinaties probeert. Daardoor neemt het aantal mogelijkheden dat geprobeerd moet worden sterk toe met het aantal kleuren is er gekozen voor een maximaal aantal kleuren van 3. Er komen ook nauwelijks werkorders voor met meer dan 3 kleuren.

## Verbeteringsheuristiek 1: andere lagencombinatie

Ook de verbeteringsheuristiek heb ik aangepast, zodat hij toepasbaar is wanneer meerdere kleuren tegelijkertijd gesneden mogen worden.

Deze verbeteringsheuristiek probeert betere oplossingen te genereren door een andere lagencombinatie te kiezen. Dit kan weer bij verschillende intekeningen gebeuren. Op één startoplossing kan de verbeteringsheuristiek dus weer meerdere keren toegepast worden.

Een door de constructieheuristiek geconstrueerde snijlijst met meerdere kleuren zit nog ingewikkelder in elkaar dan een snijlijst met één kleur, dus ook hier zorgt een kleine wijziging in een intekening er voor dat de rest van de intekeningen helemaal opnieuw bepaald moeten worden met de constructieheuristiek.

De input voor deze verbeteringsheuristiek is hetzelfde als de input voor de constructieheuristiek, plus een *startoplossing* en het *intekeningnummer* van de intekening waarbij een andere lagencombinatie gekozen moet worden. Deze heuristiek werkt op dezelfde wijze als die bij één kleur.

### Input

*behoefte(i,k)* is het aantal dat van item  $i$  en kleur  $k$  geproduceerd moet worden.

*maxlagen* is het maximaal aantal lagen dat tegelijkertijd gesneden kan worden.

*maxitems* is het maximaal aantal items dat in een intekening geplaatst mag worden.

*startoplossing* is de oplossing die verbeterd moet worden.

*intekeningnummer* is het nummer van de intekening waarbij een andere lagencombinatie gekozen moet worden.

### Stappenplan

- Stap 0:  $intekeningen = 0$ .  
 $startlagencombi(k,j)$  = het aantal lagen van dat van kleur  $k$  gesneden wordt bij intekening  $j$  in *startoplossing*.
- Stap 1:  $intekeningen = intekeningen + 1$ .  
 Als  $intekeningen < intekeningnummer$ :  
 $bestelagencombi(k) = startlagencombi(k,intekeningen)$ , voor elke kleur  $k$ , ga naar stap 4.  
 Anders:  
 $beste = 0$ . Ga naar stap 2.
- Stap 2: Ga alle mogelijke lagencombinaties af.  
 $lagencombi(k)$  is het aantal lagen van kleur  $k$ .  
 Een geldige lagencombinatie moet voldoen aan:  
 $\sum_{kleuren\ k} lagencombi(k) \leq maxlagen$ ,  
 $\sum_{kleuren\ k} lagencombi(k) > 0$ .  
 Voer stap 3 uit voor elke geldige lagencombinatie, ga daarna naar stap 4.
- Stap 3: Bepaal hoeveel items er in de intekening passen met deze lagencombinatie.  
 Voor elk item  $i$  en elke kleur  $k$ :  
 Als  $lagencombi(k) > 0$ :  
 $aantal(i,k) = \text{floor}(behoefte(i)/lagencombi(k))$ ,  
 anders:  
 $aantal(i,k) = maxitems$ .  
 Voor elk item  $i$ :  
 $aantal(i) = \min_{kleuren\ k} aantal(i,k)$ .  
 $aantal = \sum_{items\ i} aantal(i)$ .  
 Als  $aantal > maxitems$  dan  $aantal = maxitems$ .  
 Het aantal producten dat met deze intekening geproduceerd kan worden is dan:  
 $totaal = aantal * \sum_{kleuren\ k} lagencombi(k)$ .  
 Als  $intekeningen = intekeningnummer$  en voor elke kleur  $k$   $lagencombi(k) = startlagencombi(k,intekeningnummer)$ :  $totaal = 0$ .  
 Als  $totaal > beste$  dan  $beste = totaal$ ,  $bestelagencombi = lagencombi$ .
- Stap 4:  $totaalaantal = 0$ .  
 Voor elk item  $i$  en elke kleur  $k$ :  
 Als  $bestelagencombi(k) > 0$ :  
 $aantal(i,k) = \text{floor}(behoefte(i)/bestelagencombi(k))$ ,  
 anders:  
 $aantal(i,k) = maxitems$ .  
 Voor alle items  $i$ :  
 $aantal(i) = \min(maxlagen - totaalaantal, \min_{kleuren\ k} aantal(i,k))$ .  
 $totaalaantal = totaalaantal + aantal(i)$ ,  
 voor alle  $j$ :  $behoefte(i,k) = behoefte(i,k) - bestelagencombi(k) * aantal(i)$ .  
 De intekening met nummer  $intekeningen$  is nu bekend: van item  $i$  staan er  $aantal(i)$  in de intekening en van deze intekening worden  $bestelagencombi(k)$  lagen gesneden van kleur  $k$ .  
 Als voor alle items  $i$  en kleuren  $k$ :  $behoefte(i,k) = 0$ : STOP (er is een oplossing gevonden),

anders: ga naar stap 1.

### Uitleg

Net zoals bij de heuristieken voor één kleur zijn er twee belangrijke verschillen tussen de verbeteringsheuristiek en de constructieheuristiek.

1. De eerste intekeningen, tot aan de intekening die veranderd moet worden, moeten niet opnieuw bedacht worden, maar kunnen gekopieerd worden van de *startoplossing*.
2. De intekening die veranderd moet worden, moet een andere lagencombinatie hebben dan de oorspronkelijke intekening.

Het eerste punt wordt weer verzorgd in stap 1. Als het aantal intekeningen (*intekeningen*) kleiner is dan *intekeningnummer*, dan moet de intekening gekopieerd worden van *startoplossing*. De lagencombinatie wordt dan gelijk genomen aan de lagencombinatie bij deze intekening in *startoplossing* en er wordt gelijk verder gegaan met stap 4.

Het tweede verschil komt weer tot uiting in stap 3. Bij de intekening die veranderd moet worden, worden wel de stappen 2 en 3 doorlopen om de nieuwe lagencombinatie te bepalen. Echter, als het aantal intekeningen (*intekeningen*) gelijk is aan het *intekeningnummer* waarbij een andere lagencombinatie gekozen moet worden, wordt in stap 3 gekeken of de huidige lagencombinatie hetzelfde is als die van de *startoplossing*. Als dit zo is, wordt *totaal* gelijk gesteld aan 0, om zeker te zijn dat deze lagencombinatie niet gekozen zal worden.

### Het toepassen van de verbeteringsheuristiek

Deze verbeteringsheuristiek kan op dezelfde wijze toegepast worden als die bij één kleur, dus volgens het schema op bladzijde 25.

### Verbeteringsheuristiek 2: overproductie

Ook de bovenstaande heuristieken voor meerdere kleuren geven oplossingen die precies aan de vraag voldoen. Soms kan het echter gunstig zijn om iets te veel te produceren, als dit voor het snijden veel gunstiger uitkomt.

Deze heuristiek probeert om een intekening met weinig items bij een andere intekening te voegen. Eerst wordt bepaald welke intekening de minste items bevat. Vervolgens wordt bepaald bij welke intekening deze items het beste gevoegd kunnen worden. Dit moet een intekening zijn die voor elke kleur minstens evenveel lagen heeft als de intekening die bijgevoegd moet worden. Van de intekeningen die aan deze voorwaarde voldoen wordt er gekozen voor de intekening die het minste aantal lagen heeft, zodat de minste overproductie zal ontstaan. Vervolgens wordt weer gecheckt of de items wel bij deze intekening erbij passen. Als dit zo is wordt de nieuwe oplossing gemaakt.

### Input

*behoefte(i,k)* is het aantal dat van item *i* en kleur *k* geproduceerd moet worden.

*maxitems* is het maximaal aantal items dat in een intekening geplaatst mag worden.

*startoplossing* is de oplossing die verbeterd moet worden.

### Stappenplan

Stap 0: Als *startoplossing* maar 1 intekening heeft: STOP.

Stap 1: *min* is het aantal items in de intekening met de minste items,  
*minIndex* is het nummer van de intekening met de minste items,  
*minLagen(k)* = aantal lagen bij intekening met nummer *minIndex* bij kleur *k*,  
 $totaalItems = \sum_{items\ i} \sum_{kleuren\ k} behoefte(i,k)$ .

- Als  $min > (totaalItems/10)$ : STOP.
- Stap 2:  $besteAantalLagen = 1000$ ,  $besteIndex = -1$ .  
 Ga alle andere intekeningen  $j$  langs:  
 Als voor elke kleur  $k$ : aantal lagen van  $k$  bij intekening  $j > minLagen(k)$ :  
 $aantalLagen = \sum_{kleuren\ k} \text{aantal lagen van } k \text{ bij intekening } j$ ,  
 anders:  
 $aantalLagen = 1001$ .  
 Als  $aantalLagen < besteAantalLagen$ :  
 $besteAantalLagen = aantalLagen$ ,  $besteIndex = j$ .  
 Als  $besteIndex < 0$ : STOP.
- Stap 3:  $besteItem(i)$  = het aantal keer dat item  $i$  voorkomt in de intekening met nummer  $besteIndex$ ,  
 $besteItems = \sum_{items\ i} besteItem(i)$ ,  
 $minItem(i)$  = het aantal keer dat item  $i$  voorkomt in de intekening met nummer  $minIndex$ ,  
 $besteLagen(k)$  = aantal lagen van kleur  $k$  bij intekening met nummer  $besteIndex$ .  
 Voor alle items  $i$ :  
 Voor alle kleuren  $k$ :  
 $aantal(k) = \text{ceil}(minItem(i) * minLagen(k)/besteLagen(k))$ .  
 $max(i) = \text{maximum}_{kleuren\ k} aantal(k)$ .  
 $besteItemsNieuw = besteItems + \sum_{items\ i} max(i)$ .  
 Als  $besteItemsNieuw > maxItems$ : STOP.
- Stap 4: Het is mogelijk om een nieuwe oplossing te maken met een intekening minder. Alle intekeningen behalve die met nummers  $minIndex$  en  $besteIndex$  zijn hetzelfde als in *startoplossing*. Van de intekeningen met nummers  $minIndex$  en  $besteIndex$  wordt 1 intekening gemaakt, met de lagencombinatie van de intekening met nummer  $besteIndex$  ( $besteLagen$ ).  
 Voor elk item  $i$  is het aantal keer dat dit item voorkomt in deze intekening gelijk aan  $besteItem(i) + max(i)$ .

In deze heuristiek worden in de stappen 0 tot en met 3 controles gedaan of het mogelijk is om een intekening te schrappen.

In stap 0 wordt gecontroleerd of er wel meerdere intekeningen zijn. Als er maar één intekening is, kan deze niet verwijderd worden en stopt het algoritme.

In stap 1 wordt bepaald welke intekening moet verdwijnen. Dit is de intekening waarin de minste items staan.  $min$  is het aantal items dat in deze intekening zit. Dit aantal is tevens het minimum aantal overproductie dat zal ontstaan wanneer deze intekening bij een andere intekening gevoegd wordt. Als dit minimaal aantal overproductie meer dan 10% van het totaal aantal items is, dan stopt het algoritme, want meer dan 10% overproductie is te veel.

In stap 2 wordt bepaald bij welke intekening de intekening met nummer  $minIndex$  het beste gevoegd kan worden. Dit moet een intekening zijn die voor alle kleuren een hoger aantal lagen heeft dan de intekening die erbij gevoegd moet worden. En van de intekeningen die aan deze voorwaarde voldoen, moet het de intekening zijn die het minste lagen heeft.

Om deze intekening te vinden, worden eerst de variabelen  $besteAantalLagen$  en  $besteIndex$  geïnitieerd. De variabele  $besteAantalLagen$  staat voor het aantal lagen dat de beste intekening heeft,  $besteIndex$  is het nummer van deze beste intekening. Omdat gezocht wordt naar een intekening met een zo klein mogelijk aantal lagen, wordt voor  $besteAantalLagen$  in het begin een grote waarde gekozen (1000).



Vervolgens worden alle intekeningen  $j$  langsgegaan. Als een intekening aan de voorwaarde voldoet dat deze voor alle kleuren een hoger aantal lagen heeft dan de intekening die erbij gevoegd moet worden, dan wordt voor deze intekening bepaald hoeveel lagen deze heeft:  $aantalLagen$  wordt gelijkgesteld aan het aantal lagen dat deze intekening heeft ( $\sum_{kleuren\ k}$  aantal lagen van  $k$  bij intekening  $j$ ). Als deze intekening niet aan de voorwaarde voldoet, dan moet voor  $aantalLagen$  een dusdanig groot getal gekozen worden, dat deze intekening niet gekozen zal worden (1001).

Als  $aantalLagen$  kleiner is dan  $besteAantalLagen$  (de beste gevonden waarde voor  $aantalLagen$  tot dan toe), dan wordt  $besteAantalLagen$  gelijkgesteld aan  $aantalLagen$  en  $besteIndex$  wordt  $j$ . Als geen enkele intekening aan de voorwaarde voldoet, dan wordt  $aantalLagen$  steeds 1001 en dus nooit kleiner dan  $besteAantalLagen$  en dan blijft  $besteIndex$  gelijk aan -1 en wordt het algoritme beëindigd.

In stap 3 wordt gekeken of de items uit de intekening met nummer  $minIndex$  wel bij de intekening met nummer  $besteIndex$  passen. Om dit te bekijken wordt eerst bepaald hoeveel items deze intekening al bevat ( $besteItems = \sum_{items\ i} besteItem(i)$ ).

Dan wordt voor alle items bekeken hoeveel stuks van dit item extra in de intekening moeten komen om alles te produceren dat in de intekening met nummer  $minIndex$  wordt geproduceerd.

Dit wordt per kleur berekend met de formule  $aantal(k) = \text{ceil}(\text{minItem}(i) * \text{minLagen}(k) / \text{besteLagen}(k))$ . Het totaal aantal dat van item  $i$  in de intekening met nummer  $minIndex$  geproduceerd wordt is  $\text{minItem}(i) * \text{minLagen}(k)$ . Dit aantal moet in de intekening met nummer  $besteIndex$  extra geproduceerd worden. Om te weten hoe vaak dit item in de intekening erbij moet, moet dit aantal gedeeld worden door het aantal lagen dat deze intekening voor deze kleur heeft ( $\text{besteLagen}(k)$ ) en naar boven afgerond met de functie  $\text{ceil}$ .

Het aantal maal dat een item extra in de intekening moet komen ( $\text{max}(i)$ ) is het maximum van de  $aantal(k)$  van alle kleuren.

$besteItemsNieuw$  wordt het aantal items dat in de intekening zit die de twee intekeningen combineert.  $besteItemsNieuw$  wordt berekend met de formule  $besteItemsNieuw = besteItems + \sum_{items\ i} \text{max}(i)$ . Dit is het aantal items dat de intekening al bevatte ( $besteItems$ ) + het totaal aantal items dat erbij moet komen ( $\sum_{items\ i} \text{max}(i)$ ).

Als  $besteItemsNieuw$  groter is dan het maximaal aantal items dat in één intekening mag ( $\text{maxitems}$ ), dan is de nieuwe intekening dus te groot geworden. Dit is niet toegestaan en het algoritme stopt.

Als stap 4 bereikt wordt, dan is aan alle voorwaarden voldaan. Er kan een nieuwe oplossing gemaakt worden.

### **Het toepassen van de verbeteringsheuristiek**

Deze heuristiek kan worden toegepast op alle snijlijsten die met het uitvoeren van de constructieheuristiek en de eerste verbeteringsheuristiek zijn ontstaan.

### **Het bepalen van de beste oplossing**

Met alle heuristieken worden steeds nieuwe oplossingen gevonden. Sommige oplossingen hebben meer intekeningen dan andere, andere hebben weer meer overproductie. Om te bepalen welke de beste is, moet een goede criteriumfunctie gedefinieerd worden.

Eerder in dit verslag staat als criteriumfunctie vermeld: bij het maken van een snijlijst moet gezocht worden naar een snijlijst met zo min mogelijk intekeningen. Hierbij moet er wel gezorgd worden dat er niet te veel overproductie ontstaat en dat de intekeningen te veel items bevatten.

Er moet een afweging gemaakt worden tussen de drie criteria: het aantal intekeningen, het aantal overproductie en het aantal items in de intekeningen. Deze afweging kan per werkorder verschillend zijn.

Om toch enigszins de oplossingen te kunnen verdelen in betere en slechtere oplossingen heb ik de volgende criteriumfunctie vastgesteld:

$$10 * (\text{aantal intekeningen}) + (\text{aantal overproductie}) + (\text{totaal aantal items}).$$

Van de criteria is het aantal intekeningen is het belangrijkste, daarom heeft deze het gewicht 10 in de formule.

De “beste” oplossing is degene die de laagste waarde heeft voor deze formule.

## **Stofefficiëntie**

Deze criteriumfunctie houdt geen rekening met de efficiëntie van de nester. Het is van deze oplossingen ook niet bekend of ze efficiënt zijn wat stofgebruik betreft. De enige manier om hier achter te komen is door snijlijsten in te voeren in de nester.

Om een snijlijst te vinden die zuinig is met de stof en die ook goed scoort op de andere criteria, zouden van alle snijlijsten die gegenereerd zijn de snijlijsten met de beste waarden voor de criteriumfunctie kunnen worden ingevoerd in de nester. Daarna kan aan de hand van de resultaten worden bepaald welke snijlijst gebruikt gaat worden bij de productie.

## **Zwakke punten van de heuristieken**

Deze heuristieken zijn gemaakt om snel goede snijlijsten te genereren. Ze geven echter niet gegarandeerd de optimale oplossing.

Een reden hiervoor is dat de heuristieken niet vooruit kijken. Ze maken één voor één intekeningen die op dat moment het beste zijn, zonder te kijken wat dit betekent voor de behoefte die je daarna overhoudt.

Dit wordt deels opgelost door het steeds weer toepassen van de verbeteringsheuristiek. Daardoor worden meerdere aantallen lagen uitgetoetst en is de kans dat een betere oplossing gevonden wordt groter.

Toch lost het steeds weer toepassen van de verbeteringsheuristiek niet het hele probleem van het niet vooruitkijken op. Stel dat van een bepaald item 18 stuks nodig zijn, en er is een snijlijst gemaakt met 4 intekeningen, met 13, 9, 2 en 1 lagen. Dan zal door de heuristieken altijd het betreffende item 1 keer in de intekening met 13 lagen komen, vervolgens moeten er dan nog 5 geproduceerd worden. Dan wordt dit item nog 2 keer in de intekening met 2 lagen gestopt en 1 keer in de intekening met 1 laag. Dit zorgt ervoor dat dit item in totaal 4 keer uitgesneden moet worden. Maar als vooruit was gekeken was dit item niet in de intekening met 13 lagen gestopt, maar 2 keer in de intekening met 9 lagen en dan hoefde het maar 2 keer uitgesneden te worden.

Een probleem van de constructieheuristiek is dat de intekeningen vaak helemaal vol zitten met het maximale aantal items. Dit komt doordat deze heuristiek erop gebaseerd is om zo veel mogelijk items in een keer te snijden. Maar volle intekeningen kosten veel tijd om te snijden. Ook dit probleem wordt deels opgelost door de verbeteringsheuristiek vaak toe te passen. Zo worden ook hogere aantallen lagen geprobeerd en worden ook oplossingen met minder volle intekeningen gevonden.

Een zwak punt van de overproductieheuristiek is dat deze altijd op een oplossing moet worden toegepast en niet verweven is met de constructie heuristiek. Wanneer bij het construeren van een oplossing al rekening wordt gehouden met de mogelijkheid van overproductie, zouden wellicht nog beter oplossingen gevonden kunnen worden.

Een nadeel van de heuristieken met meerdere kleuren is dat alle lagencombinaties geprobeerd worden. Daardoor neemt het aantal mogelijkheden dat geprobeerd moet worden sterk toe met het aantal kleuren en het maximaal aantal lagen en is er gekozen voor een maximaal aantal kleuren van 3. Er komen ook nauwelijks werkorders voor met meer dan 3 kleuren.

De heuristieken houden geen rekening met het efficiënt gebruiken van de stof. Dit is onmogelijk door de lange rekentijd die de nester heeft. Uit het onderzoek dat eerder met de nester gedaan is (zie bijlage A) bleek dat wanneer minimaal vijf items in een intekening zitten, de stof in ieder geval redelijk efficiënt gebruikt wordt. Ook dit minimum van vijf wordt niet meegenomen in de heuristieken. Wel is het zo dat bij het zoeken naar een snijlijst met zo min mogelijk intekeningen, automatisch de intekeningen meestal wel meer dan vijf items bevatten.



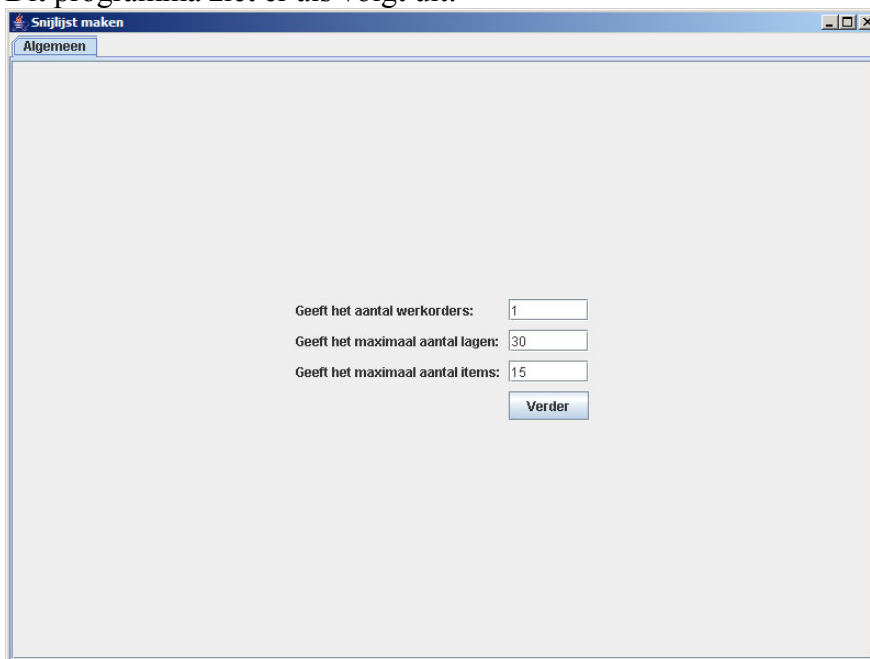
## Hoofdstuk 4 Implementatie en tests

In dit hoofdstuk staat hoe de heuristieken zijn geïmplementeerd. Er wordt aan de hand van screenshots uitgelegd hoe het programma werkt. In de tweede paragraaf worden testresultaten gegeven. Er worden een aantal oude werkorders getest en de resultaten die bereikt worden met het programma worden vergeleken met de resultaten van de huidige methode. Ook wordt de snelheid van het programma getest. Tenslotte wordt nog uitgelegd wat er gebeurt als producten uit verschillende stofsoorten bestaan.

### *Het programma*

Om de heuristieken goed te kunnen testen zijn ze geïmplementeerd. Er is een programma gemaakt in JAVA. Dit is een op zichzelf staand programma. In dit programma kunnen gegevens worden ingevuld van één of meerdere werkorders en dit programma past dan de heuristieken toe en geeft snijlijsten als resultaat.

Dit programma ziet er als volgt uit:



In dit eerste scherm moeten 3 waarden worden ingevuld.

Het aantal werkorders dat gecombineerd moet worden, het maximaal aantal lagen stof dat tegelijkertijd gesneden kan worden en het maximaal aantal items dat in een intekening mag komen.

Vervolgens worden tabbladen gemaakt voor de werkorders. Het aantal tabbladen dat aangemaakt wordt is gelijk aan het ingevulde aantal werkorders.

Als de input niet klopt, bijvoorbeeld negatieve getallen of letters, dan wordt een foutmelding gegeven.

Naam werkorder

Aantal maten

Aantal cupmaten

Aantal kleuren (typen)

Vorige Volgende

Per tabblad moeten de gegevens van een werkorder ingevuld worden. Elk tabblad bestaat uit 3 schermpjes.

In het eerste schermpje moeten wat gegevens over de werkorder worden ingevuld: de naam van de werkorder, het aantal maten, het aantal cupmaten en het aantal kleuren dat voorkomt op deze werkorder.

Naam werkorder

Aantal maten

Aantal cupmaten

Aantal kleuren (typen)

Vorige Volgende

Als op de knop “Volgende” wordt gedrukt verschijnt het volgende scherm.

In dit scherm kunnen de namen van de maten, cupmaten en kleuren worden ingevuld. Standaard zijn hier al wat waarden ingevuld, zodat bij de meest voorkomende maten niet steeds alles ingevuld hoeft te worden.

The screenshot shows a window titled 'Snijlijst maken' with a tab labeled 'WERKORDER1'. The window contains the following input fields:

- Maten:** A row of five input boxes containing the values 38, 40, 42, 44, and 46.
- Cup maten:** A single input box containing the value 'B'.
- Kleuren:** A single input box containing the value '991'.

At the bottom of the input section, there are two buttons: 'Vorige' and 'Volgende'.

Als weer op de knop “Volgende” wordt gedrukt verschijnt het laatste scherm van deze werkorder. Op dit scherm staat een tabel waarin de behoefte van de werkorder moet worden ingevuld. Als kolomnamen zijn de maten genomen die ingevuld zijn in het vorige scherm. En voor elke kleur en cupmaat combinatie is een rij gemaakt.

Zo kan overzichtelijk de behoefte ingevuld worden.

The screenshot shows the same window 'Snijlijst maken' with the 'WERKORDER1' tab. The main area contains the following text and table:

Vul hier de behoefte van werkorder 3943 in

Kleur	Cup	38	40	42	44	46
691	B	36	68	98	86	44
991	B	22	54	80	68	48

At the bottom, there are two buttons: 'Vorige' and 'Maak snijlijst'.

Als dit de enige of laatste werkorder is, dan is er ook een knop “Maak snijlijst”. Als er nog een volgende werkorder is, is er een knop “Volgende”, waarmee het eerste scherm van de volgende werkorder bereikt wordt.

Als op de knop “Maak snijlijst” gedrukt wordt, dan worden alle waarden uit de invulvelden ingelezen en worden de heuristische toegepast. Er worden met deze heuristieken meerdere snijlijsten gemaakt. Voor alle snijlijsten wordt de waarde voor de criteriumfunctie bepaald en op basis daarvan worden de snijlijsten gesorteerd.

Er verschijnt een tabblad met het resultaat. Op dit tabblad staat de beste gevonden snijlijst volgens de criteriumfunctie weergegeven in een tabel.

Ook staat boven de tabel hoeveel intekeningen deze snijlijst bevat, hoeveel overproductie er is en hoeveel items uitgesneden moeten worden.

**Oplossing 1 van de 367. Deze oplossing heeft 4 intekeningen, 2 overproductie en 33 items uit te snijden.**

Intekening	Werkorder	Cup	38	40	42	44	46	Lagen 691	Lagen 991
1	3943	B	1	4	5	5	0	17	13
2	3943	B	1	0	2	0	7	6	6
3	3943	B	2	0	0	0	0	6	0
4	3943	B	1	1	1	1	2	1	3

Volgende oplossing

Met de knop “Volgende oplossing” kan de snijlijst weergegeven worden die het op één na beste scoort bij de criteriumfunctie.

**Oplossing 2 van de 367. Deze oplossing heeft 3 intekeningen, 16 overproductie en 30 items uit te snijden.**

Intekening	Werkorder	Cup	38	40	42	44	46	Lagen 691	Lagen 991
1	3943	B	1	3	5	4	2	18	12
2	3943	B	1	2	2	2	3	4	10
3	3943	B	3	1	0	1	0	6	0

Vorige oplossing    Volgende oplossing

Wanneer op de knop “Volgende oplossing” wordt geklikt, dan wordt de volgende snijlijst weergegeven. Met de knop “Vorige oplossing” kan terug worden gegaan naar de vorige snijlijst. Zo kunnen eenvoudig de snijlijsten bekeken worden en kan men ze met elkaar vergelijken.

De gekozen snijlijst zal vervolgens per intekening in de nester ingevoerd moeten worden om het stofverbruik te bepalen.



## Testresultaten

Om een beeld te krijgen van hoe goed de heuristieken werken, zijn ze toegepast op een aantal oude werkorders. Van deze werkorders worden de snijlijst die het huidige programma voor het maken van snijlijsten geeft of de aangepaste snijlijst, die daadwerkelijk gebruikt is, vergeleken met de nieuwe snijlijsten die het JAVA-programma genereert.

### Werkorder 5122

De eerste werkorder die getest wordt is werkorder 5122. Dit is een werkorder van deze bikini:



Deze bikini wordt in 5 maten, 3 cupmaten en 2 kleuren (typen) gemaakt.

Behoefte WO

Type	Cup	36	38	40	42	44
390	B	48	67	63	28	7
390	C	11	41	41	29	4
390	D	9	10	16	11	4
661	B	23	34	35	18	7
661	C	11	24	27	17	2
661	D	5	10	11	10	4

Het programma dat nu de snijlijsten creëert maakt voor deze werkorder de volgende snijlijst:

Snijlijst

Intekening	WO	Cup	36	38	40	42	44	lagen 390	lagen 661
1	5122	B	4	5	5	2	0	12	
2	5122	B	0	3	1	2	3	2	
3	5122	B	0	1	1	0	1	2	
4	5122	C	1	5	5	3	0	8	
5	5122	C	2	1	1	3	2	2	
6	5122	D	3	3	5	3	1	3	
7	5122	D	0	1	1	1	1	2	
8	5122	B	3	4	5	2	1		7
9	5122	B	1	3	0	2	0		2
10	5122	C	1	3	3	2	0		8
11	5122	C	2	0	2	1	1		2
12	5122	D	2	5	5	5	2		2
13	5122	D	1	0	1	0	0		2

Deze snijlijst heeft 13 intekeningen; er moeten in totaal 127 items uitgesneden worden; er is een overproductie van 15 items.

Deze snijlijst is met de hand aangepast en zo is de volgende snijlijst tot stand gekomen. Dit is de snijlijst die gebruikt is bij de productie:

Snijlijst

Intekening	WO	Cup	36	38	40	42	44	lagen 390	lagen 661	Meter
1	5122	B	4	5	5	2	0	12		4,27
2	5122	B	0	4	2	2	4	2		6,1
2	5122	C	2	1	1	3	2	2		
3	5122	C	1	5	5	3	0	8		3,94
4	5122	D	2	3	4	3	1	4		3,86
5	5122	B	2	3	3	1	0		11	4,07
5	5122	C	1	2	2	1	0		11	
6	5122	D	1	2	2	2	0		5	2,07
7	5122	B	1	1	1	3	3		2	6,27
7	5122	C	1	1	3	3	1		2	
7	5122	D	0	0	1	0	2		2	

Deze snijlijst heeft 7 intekeningen; er moeten in totaal 107 items uitgesneden worden; er is een overproductie van 13 items.

Deze aangepaste snijlijst scoort op alle criteria beter dan de oorspronkelijke snijlijst. Deze snijlijst is ingevoerd in de nester. Bij elke intekening in de tabel staat in de laatste kolom de lengte van deze intekening. De totale hoeveelheid stof die gebruikt wordt kan berekend worden door voor elke intekening het aantal lagen te vermenigvuldigen met de lengte van de intekening. Bij de lengte van de intekening moet wel steeds 4 centimeter opgeteld worden, omdat aan beide kanten 2 centimeter stof extra genomen moet worden. Deze aantallen bij elkaar opgeteld is de hoeveelheid stof die verbruikt wordt: 179,82 meter.

De gegevens van deze werkorder zijn ingevoerd in het JAVA-programma. Als maximaal aantal lagen is 30 gekozen en als maximaal aantal items per intekening is 20 gekozen.

De eerste oplossing die gegeven wordt is:

Snijlijst

Intekening	WO	Cup	36	38	40	42	44	lagen 390	lagen 661	Meter
1	5122	B	2	3	3	1	0	19	11	3,82
1	5122	C	0	2	2	1	0	19	11	
2	5122	B	0	0	0	1	1	5	5	3,53
2	5122	C	2	0	0	1	0	5	5	
2	5122	D	1	2	2	2	0	5	5	
3	5122	B	2	2	1	1	0	4	0	3,13
3	5122	C	0	0	0	1	1	4	0	
3	5122	D	1	0	1	0	1	4	0	
4	5122	B	0	0	0	1	0	0	2	1,92
4	5122	C	0	0	1	0	1	0	2	
4	5122	D	0	0	0	0	2	0	2	
5	5122	B	2	2	2	0	2	1	1	5,35
5	5122	C	1	3	3	1	0	1	1	
5	5122	D	0	0	2	1	0	1	1	

Deze snijlijst heeft 5 intekeningen; er moeten in totaal 61 items uitgesneden worden; er is een overproductie van 5 items.

Op alle criteria scoort deze snijlijst beter dan de met de hand aangepaste snijlijst. Ook deze snijlijst is ingevoerd in de nester en de lengtes van de intekeningen zijn in de tabel vermeld.

De totale hoeveelheid stof die verbruikt wordt is 178,88 meter.

Het programma geeft voor deze werkorder nog een groot aantal alternatieve oplossingen. Een daarvan bevat slechts 4 intekeningen, deze heeft echter wel 19 items overproductie. Ook wordt een oplossing met helemaal geen overproductie gevonden, deze heeft echter wel 6 intekeningen. Er is dus een groot aantal mogelijke snijlijsten waar uit gekozen kan worden.

Om te bepalen hoeveel voordeel het samenvoegen van de kleuren oplevert, is deze werkorder ook getest voor de beide kleuren apart. Om dit te doen is twee keer het JAVA-programma gedraaid, de eerste keer voor kleur 390 en de tweede keer voor kleur 661.

Dit geeft het volgende resultaat:

#### Snijlijst

Intekening	WO	Cup	36	38	40	42	44	lagen	lagen	Meter
								390	661	
1	5122	B	1	2	2	1	0	27		2,49
1	5122	C	0	1	1	1	0	27		
2	5122	B	3	1	1	0	1	7		4,42
2	5122	C	1	2	2	0	0	7		
2	5122	D	1	1	2	1	0	7		
3	5122	B	0	3	1	1	0	2		5,24
3	5122	C	2	0	0	1	2	2		
3	5122	D	1	2	1	2	2	2		
4	5122	B	1	2	2	1	0		17	2,49
4	5122	C	0	1	1	1	0		17	
5	5122	B	1	0	0	0	1		5	4,01
5	5122	C	2	1	2	0	0		5	
5	5122	D	1	2	2	2	0		5	
6	5122	B	1	0	1	1	2		1	4,6
6	5122	C	1	2	0	0	2		1	
6	5122	D	0	0	1	0	4		1	

Deze snijlijst bevat 6 intekeningen; er moeten in totaal 81 items uitgesneden worden; er is een overproductie van 2 items. De totale hoeveelheid stof die verbruikt wordt is 178,88 meter.

#### Overzicht resultaten werkorder 5122:

	aantal intekeningen	aantal overproductie	aantal items uitsnijden	aantal meter stof
Systeem oplossing	13	15	127	
Verbeterde systeem oplossing	7	13	107	179,82
Nieuwe oplossing	5	5	61	178,88
Nieuwe oplossing met kleuren apart	6	2	71	177,99

De heuristieken blijken goed te werken, want de nieuwe oplossingen hebben ruim 50% minder intekeningen, overproductie en items uit te snijden dan de oplossing die het oude programma maakt. Ook zijn de nieuwe oplossingen beter dan de met de hand aangepaste snijlijst.

De nieuwe oplossing met de kleuren samengevoegd is beter dan de nieuwe oplossing met kleuren apart.

De hoeveelheid verbruikte stof is minder bij de twee nieuwe oplossingen dan bij de oude. Deze vergelijking is niet helemaal eerlijk, omdat bij de oude snijlijst meer overproductie is.

Wanneer het aantal meter gebruikte stof wordt omgerekend naar het aantal centimeter stof per geproduceerd item, dan blijkt dit bij alle drie de snijlijsten 28 centimeter te zijn. Het stofverbruik is dus niet verbeterd.

### Werkorder 4163

Een andere werkorder die getest is, is werkorder 4163. Dit is de werkorder van een badpak dat in 5 maten, 2 cupmaten en 3 kleuren (typen) gemaakt wordt. Deze werkorder is op dezelfde wijze getest als werkorder 5122. Als maximaal aantal lagen is hierbij weer 30 gekozen en als maximaal aantal items per intekening is 15 gekozen.

De tabellen van de behoefte en snijlijsten staan in bijlage B.

Overzicht resultaten werkorder 4163:

	aantal intekeningen	aantal overproductie	aantal items uitsnijden	aantal meter stof
Systeem oplossing	14	15	119	533,16
Nieuwe oplossing	6	7	74	531,95
Nieuwe oplossing met kleuren apart	9	3	79	535,86

Deze resultaten zijn ongeveer hetzelfde als de resultaten van werkorder 5122. Er is een vermindering van ruim 50% behaald in het aantal intekeningen, overproductie en items.

De oplossing met de kleuren samengevoegd is beter dan de oplossing met de kleuren apart.

Bij de hoeveelheid benodigde stof is her wel iets opvallends te zien. De nieuwe oplossing met kleuren apart kost meer stof per item dan de andere twee oplossingen. Dit komt doordat deze oplossing een intekening bevat met slechts 2 items en van deze intekening worden wel 18 lagen gesneden.

### Werkorders 5102 en 5158

De volgende werkorders die getest zijn, zijn de werkorders 5102 en 5158. Werkorder 5102 is de werkorder van een bikini die in 5 maten en 3 cupmaten gemaakt moet worden. Werkorder 5158 is de werkorder van een rok die in 5 maten gemaakt moet worden. Deze rok en bikini bestaan uit dezelfde stof, dus deze werkorders kunnen worden samengevoegd om zo een efficiëntere snijlijst te maken.

Deze werkorders zijn bij de productie samengevoegd. Toen is met de hand de snijlijst voor deze werkorders gemaakt.

Deze werkorders zijn samen ingevoerd in het JAVA-programma. Als maximaal aantal lagen is 30 gekozen en als maximaal aantal items per intekening is 20 gekozen.

Ook zijn deze werkorders apart ingevoerd in het JAVA-programma.

De tabellen van de behoefte en de snijlijsten zijn weer te vinden in bijlage B.

Overzicht resultaten werkorders 5102 en 5158:

	aantal intekeningen	aantal overproductie	aantal items uitsnijden	aantal meter stof
Systeem oplossing werkorder 5102	7	3	40	
Handmatige oplossing werkorders samen	4	2	50	119,14
Nieuwe oplossing 1	3	14	42	128,34
Nieuwe oplossing 2	4	0	47	119,09
Nieuwe oplossing met werkorders apart 1	4	10	47	137,32
Nieuwe oplossing met werkorders apart 2	5	0	54	133

Bij deze werkorders zijn de resultaten ook goed. De oplossing die het systeem geeft voor werkorder 5102 alleen heeft al 2 keer zo veel intekeningen dan de nieuwe oplossingen voor de twee werkorders gecombineerd.

De nieuwe oplossingen zijn net iets beter dan de handmatige oplossing.

De nieuwe oplossingen met de twee werkorders apart zijn net iets minder dan de handmatige oplossing. Het grote nadeel van de oplossingen voor de twee werkorders apart is dat ze veel meer stof kosten dan de andere oplossingen. Dit komt doordat de patroondelen van de rok van werkorder 5158 vrij groot zijn en slecht in elkaar passen. Het is dus voornamelijk voor het stofgebruik gunstig om deze werkorder 5158 te combineren met werkorder 5102 van een bikini met kleinere patroondelen.

## **De snelheid van het programma**

Het programma werkt zeer snel. Bij de geteste werkorders waren telkens binnen enkele seconden alle heuristieken uitgevoerd.

## **Conclusies**

- Het JAVA-programma maakt snijlijsten met ongeveer 50% minder intekeningen, overproductie en items uit te snijden dan het oude programma.
- De snijlijsten die het JAVA-programma maakt zijn meestal net iets beter dan een handmatig gemaakte snijlijst.
- Het stofverbruik wordt niet verbeterd met het nieuwe programma, maar is vergelijkbaar met het stofverbruik van het oude programma.
- Het combineren van kleuren en werkorders levert efficiëntere snijlijsten.

## **Meerdere stofsoorten**

In de praktijk bestaan de meeste producten uit meerdere soorten stof, zoals bovenstof en voering. Dit staat los van het feit dat sommige producten in verschillende kleuren geproduceerd worden. Het programma maakt een snijlijst voor een product dat maar uit één soort stof bestaat. Dit is gelukkig geen probleem, want voor elke stof van een product kan het programma opnieuw uitgevoerd worden, en zo wordt voor elke stofsoort een aparte snijlijst gemaakt.

Stel dat een badpak bestaat uit twee soorten hoofdstof (A-stof en B-stof) en voering. Bijvoorbeeld: grootste deel van het badpak is gemaakt van de A-stof, maar er zit ook een streep op het badpak van een andere kleur, de B-stof. En er zit ook nog voering in het badpak. Dan kan voor dit product een snijlijst gemaakt worden door het programma drie keer uit te voeren.

Er moeten drie keer bijna dezelfde waarden in het programma ingevuld worden, toch kunnen er wel wat verschillen zijn. Bijvoorbeeld, bij de B-stof kan voor het maximaal aantal items in een intekening een hoog aantal ingevuld worden, want de patroondelen voor de B-stof zijn klein. Voor de voering zou een hoger maximaal aantal lagen ingevuld kunnen worden, want de voering is dunnere stof. De snijlijsten die het programma voor de drie stofsoorten geeft, vormen samen de snijlijst voor dit product.

Stel dat dit badpak ook nog in meerdere kleurencombinaties (typen) gemaakt wordt, dan kunnen deze gecombineerd worden met de heuristieken voor meerdere kleuren. Dan kan voor de A-stof en B-stof het programma op dezelfde wijze uitgevoerd worden, maar dan met meerder kleuren. Voor de voering kan dan het aantal benodigde items van de verschillende kleurencombinaties opgeteld worden, want de voering is voor de verschillende typen badpakken hetzelfde.

Een probleem dat hierbij ontstaat is de overproductie. Wanneer voor elke stofsoort een andere snijlijst met andere overproductie gemaakt wordt, kunnen problemen ontstaan met het in elkaar naaien van overproductie. Dit is bijvoorbeeld het geval als van een item wel overproductie is van de A-stof, maar niet van de B-stof of de voering.

Dit probleem kan op twee manieren opgelost worden.

- Er kan voor gekozen worden om de overproductie als stofverlies te zien en de te veel geproduceerde items die niet compleet zijn moeten dan worden weggegooid. In dit geval moeten snijlijsten gekozen worden met weinig overproductie.
- Een andere keuze is om eerst van de duurste stof een snijlijst te maken. Dan is van deze stof de overproductie bekend. Vervolgens kan voor de items die teveel geproduceerd worden de behoefte aangepast worden en de snijlijst voor de volgende stof gemaakt worden.

## **Hoofdstuk 5 Conclusies en aanbevelingen**

### ***Conclusies***

Het proces van het maken van snijlijsten en intekeningen is erg ingewikkeld. Er zijn ontzettend veel mogelijkheden en ook erg veel dingen waar rekening mee moet worden gehouden.

Het is niet precies duidelijk welke snijlijsten het beste zijn. Dit is altijd een afweging tussen verschillende factoren.

Voor de snelheid van het snijproces is het van belang om het aantal intekeningen te minimaliseren. Daarmee wordt bereikt dat een minimaal aantal malen het tijdrovende proces van lagen stof opgespannen moet worden doorlopen. Ook is het goed als de intekeningen niet te veel items bevatten, zodat het uitsnijden niet te veel tijd gaat kosten.

Voor de stoffefficiëntie is het van belang dat elke intekening minimaal vijf items bevat. Ook is het efficiënt om verschillende producten en maten in een intekening te mixen.

Het gegarandeerd vinden van de ‘optimale’ snijlijst in redelijke tijd is onmogelijk; er zijn te veel mogelijkheden en er is geen oplossingsmethode voor handen die daaruit gegarandeerd de beste kiest.

Voor het snel vinden van een goede snijlijsten zijn verschillende heuristische ontwikkeld. Deze heuristieken maken het mogelijk om verschillende cupmaten en werkorders met elkaar te combineren. Ook bieden zij de mogelijkheid om verschillende kleuren stof tegelijkertijd te snijden.

Met deze heuristieken zijn grote verbeteringen behaald met de helft minder intekeningen, overproductie en items die uitgesneden moeten worden ten opzichte van de oude methode. Ook bij het niet samenvoegen van werkorders en kleuren worden veel betere oplossingen gevonden.

Voor een efficiënt snijproces met zo min mogelijk intekeningen, zo min mogelijk overproductie en zo min mogelijk items die uitgesneden moeten worden is het gunstig om zoveel mogelijk te combineren. Hoe meer werkorders en kleuren gecombineerd worden, hoe efficiënter de snijlijsten worden.

Het stof gebruik is niet verbeterd, dit was ook niet te verwachten, omdat hier niet op is geoptimaliseerd. Wel is het waarschijnlijk zo dat mixen van verschillende werkorders gunstig is voor het stofverbruik.

### ***Aanbevelingen***

Een van de conclusies luidt dat het efficiënt is om zo veel mogelijk werkorders en kleuren te combineren. Echter, in de praktijk waarschijnlijk is dit waarschijnlijk onhandig, omdat na het uitsnijden de patroondelen van de items op de juiste manier bij elkaar gezocht moeten worden, om ze vervolgens aan elkaar te naaien. Als alle werkorders en kleuren door elkaar gesneden worden kan het bij elkaar zoeken van de juiste patroondelen een lastige taak worden, die weer veel tijd in beslag gaat nemen, of waar fouten bij gemaakt worden.

Daarom is het waarschijnlijk beter om niet altijd alles wat gecombineerd kan worden te combineren. Er dient hier een middenweg gevonden te worden.

Een reden om werkorders samen te voegen is wanneer bij een bepaalde werkorder de patroondelen van dit product niet zo mooi in elkaar te leggen zijn. Dan is het gunstig voor het stofverbruik deze patroondelen te mengen met patroondelen van een ander product.

Een andere reden om werkorders samen te voegen is wanneer de werkorders klein zijn. Wanneer een snijlijst gemaakt wordt voor te weinig items, dan is deze meestal minder efficiënt dan een snijlijst voor heel veel items.

In het uiteindelijke systeem moet het proces van snijden helemaal automatisch gaan. In het systeem moeten meerdere werkorders geselecteerd kunnen worden. Dan moet het systeem van de PIS-formulieren die horen bij deze werkorders de benodigde informatie verzamelen.

Benodigde informatie van de werkorders en PIS-formulieren:

- het aantal dat van elk item geproduceerd moet worden,
- uit welke stoffen en patroondelen de producten bestaan (bijvoorbeeld: A-stof, B-stof, voering),
- het aantal centimeter stof dat van elke stof gemiddeld per product gebruikt wordt.
- het maximaal aantal lagen dat tegelijkertijd gesneden kan worden van elke stof.

Dan moet per stofsoort bepaald worden hoeveel er precies moet worden uitgesneden van welke patroondelen, en hoeveel items er maximaal in een intekening mogen komen.

Dit gebeurt al in het huidige programma.

Met deze informatie kan per stof een snijlijst bepaald worden met de hierboven beschreven heuristieken. Per stof kan een programma dat ongeveer gelijk is aan het door mij geschreven JAVA-programma aangeroepen worden. Dit programma creëert meerdere mogelijke oplossingen, sorteert de oplossingen op kwaliteit en geeft deze terug. Er moet vervolgens door de gebruiker handmatig gekozen kunnen worden uit de gegeven oplossingen.

De snijlijsten van alle stoffen samengevoegd vormt de snijlijst voor deze werkorders. Vervolgens moet deze snijlijst automatisch ingevoerd kunnen worden in de nester.



# Bijlage A: Nesteronderzoek

## Inleiding

De nester heeft veel tijd nodig om te rekenen, en daarom kan de nester niet tijdens het optimaliseren van de snijlijst gebruikt worden. Om toch enig inzicht te krijgen in de stoffefficiëntie van de nester, heb ik dit onderzoek gedaan naar efficiënt stofgebruik.

Met dit onderzoek dienen de volgende vragen te worden beantwoord:

1. Hoeveel items moeten er minimaal worden ingelegd, zodat de hele stofbreedte benut wordt en er redelijk efficiënt met de stof omgegaan wordt?
2. Is het voordelig of juist onvoordelig voor het stofgebruik om verschillende werkorders samen te voegen, dus is het voordelig om verschillende soorten producten in een intekening te plaatsen?

## Onderzoek 1

### Onderzoeksvraag

Hoeveel items moeten er minimaal worden ingelegd, zodat de hele stofbreedte benut wordt en er redelijk efficiënt met de stof omgegaan wordt?

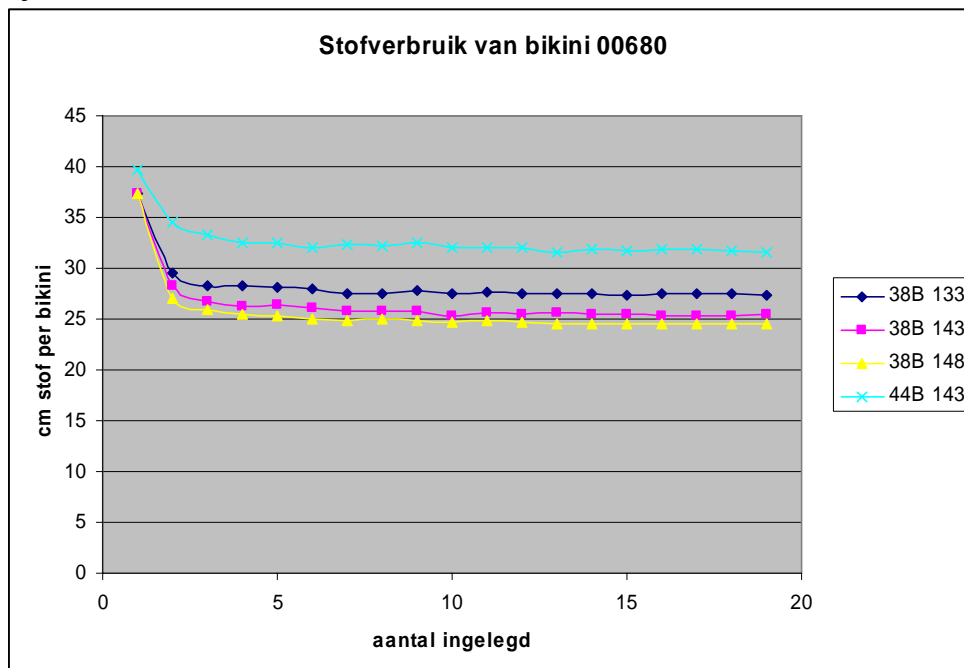
### Werkwijze

Het stofverbruik bij het inleggen van verschillende aantallen van dezelfde producten meten, bij verschillende maten en stofbreedtes.

Dit onderzoek is uitgevoerd voor bikini 680, badpak 61-0803 en jurk 1504.

### Resultaat

Dit is een grafiekje van het stofverbruik van de bikini, als je verschillende aantallen inlegt. Er zijn 4 reeksen, voor verschillende stofbreedtes en maten.

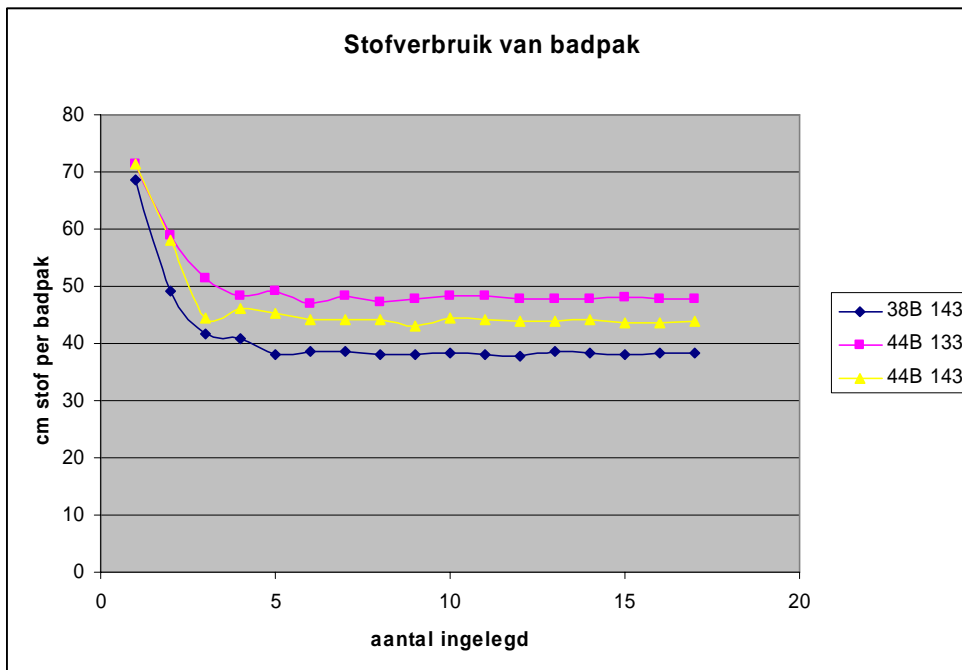


De bovenste reeks is voor maat 44B en stofbreedte 143. Het eerste punt van de grafiek is het aantal centimeter stof dat nodig is voor het uitsnijden van 1 bikini maat 44B. Dit is bijna 40 centimeter. Als 2 bikini's van deze maat uitgesneden worden, dan kost dit ongeveer 70

centimeter stof. Dit komt overeen met 35 centimeter per bikini, en dat is het tweede punt van de grafiek. Dit is dus een stuk efficiënter.

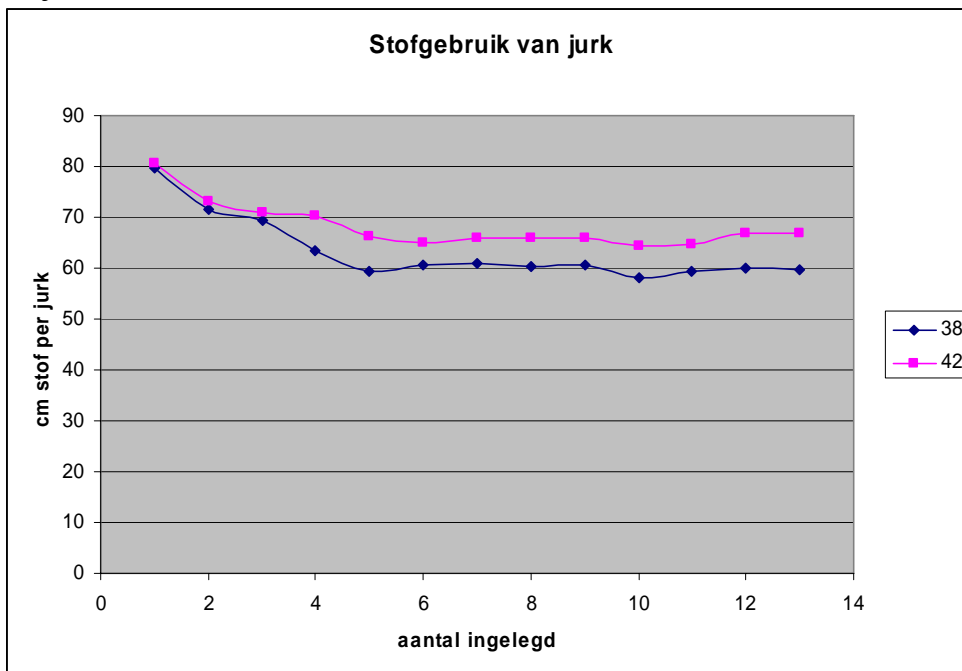
Alle 4 de grafieken hebben ongeveer hetzelfde verloop. Als slechts 1 bikini wordt uitgesneden, wordt de stof niet helemaal benut, en is het stofgebruik inefficiënt. Vanaf ongeveer 4 of 5 ingelegde bikini's dalen de lijnen niet meer, dit houdt in dat het voor het stofgebruik per bikini niet uitmaakt of er 5, 10 of 15 worden ingelegd.

Hieronder staat dezelfde grafiek voor het badpak.



Ook hier valt dezelfde trend te zien. Het stofgebruik per badpak daalt totdat er 4 of 5 zijn ingelegd, daarna is het stofgebruik per badpak constant.

De jurk:



Deze grafiek is iets grilliger dan die van de bikini en het badpak. Waarschijnlijk komt dit doordat de wat grotere patroondelen van de jurk soms juist goed en soms juist slecht in elkaar

kunnen passen. Maar ook hier is te zien dat er vanaf 5 jurken per intekening, efficiënt met de stof wordt omgegaan.

## Conclusie onderzoek 1

vijf is een goed minimum aantal items per intekening, zodat er efficiënt met de stof om wordt gegaan.

## Onderzoek 2

### Onderzoeksvraag

Is het voordelig of juist onvoordelig voor het stofgebruik om verschillende werkorders samen te voegen, dus is er stofvoordeel te behalen om verschillende soorten producten in een intekening te plaatsen?

### Werkwijze

Het stofverbruik is gemeten bij intekeningen waarbij verschillende producten door elkaar heen zijn gelegd en vergeleken met het stofgebruik van die producten apart.

Dit onderzoek is weer uitgevoerd met bikini 680, badpak 61-0803 en jurk 1504. Steeds zijn in totaal 12 producten in een intekening gezet, maar dan steeds andere aantallen voor jurk, badpak en bikini. Dus bijvoorbeeld 3 badpakken, 4 bikini's en 5 jurken, of 1 badpak, 1 bikini en 10 jurken.

In totaal zijn 46 combinaties getest.

### Resultaat

Uit onderzoek 1 komt dat bij stofbreedte 143, de bikini maat 38B in het gunstigste geval (bij het inleggen van 10 bikini's) 25,037 cm stof per bikini kost.

Voor het badpak maat 38B is dit 37,615 cm (bij een intekening van 12 badpakken).

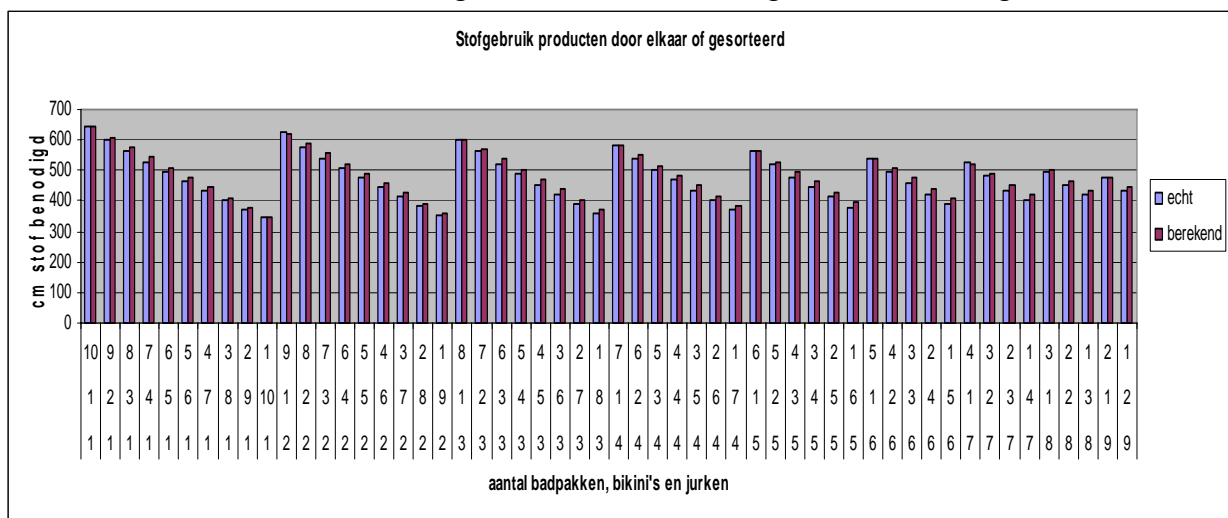
Voor de jurk maat 38 is dit 57,99 cm (bij een intekening van 10 jurken).

Voor elke gemaakte intekening met verschillende producten is uitgerekend hoeveel stof deze intekening zou moeten kosten met de formule:

$(\text{aantal badpakken}) * 37.615 + (\text{aantal bikini's}) * 25.037 + (\text{aantal jurken}) * 57.99.$

Dit getal is vervolgens vergeleken met lengte van de intekening in werkelijkheid.

Hieronder staat een staafdiagram, met per intekening het aantal cm benodigde stof en de hoeveelheid stof die deze intekening zou moeten kosten volgens de berekening.



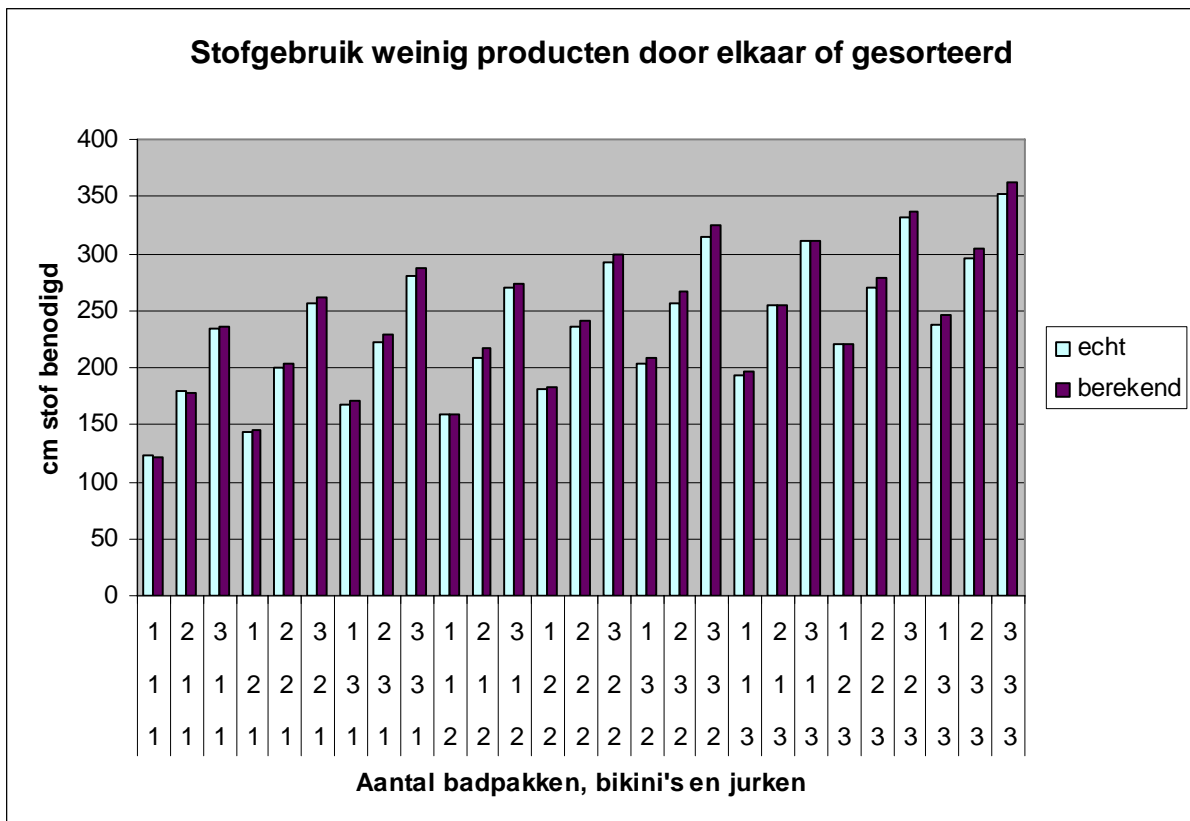
Te zien is dat meestal het daadwerkelijke stofverbruik lager is dan het berekende stofverbruik. Gemiddeld is het stofverbruik bij producten door elkaar in 1 intekening geplaatst meer dan 10 cm minder per intekening dan wanneer de producten gesorteerd worden geproduceerd.

## Conclusie onderzoek 2

Wanneer verschillende producten in een intekeningen worden gemixt, kost een product gemiddeld meestal minder stof dan wanneer intekeningen maar één product bevatten.

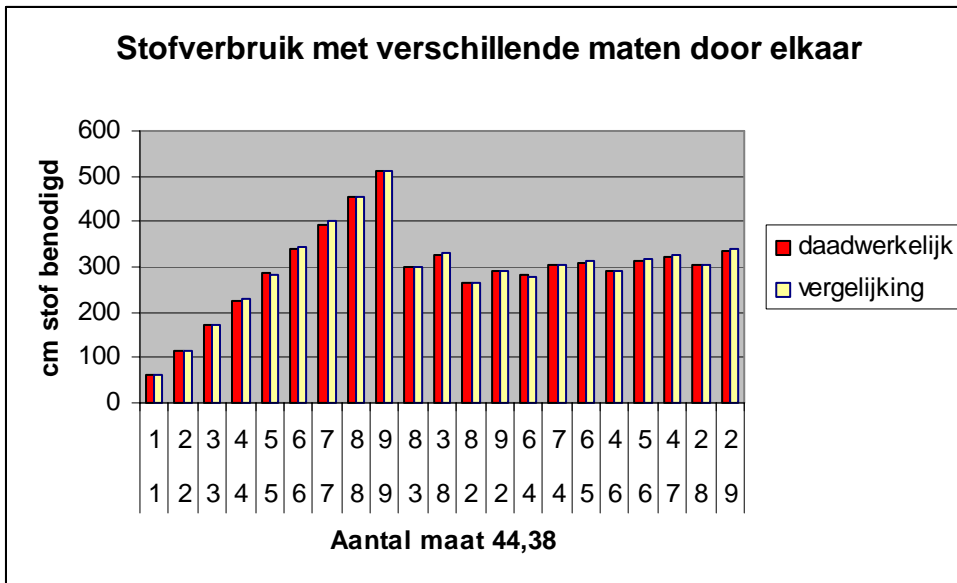
### Extra onderzoek

Omdat het gebruik van verschillende producten in een intekening voordelig is, is ook getest of het minimaal aantal producten in een intekening van vijf ook genoeg is wanneer verschillende producten in een intekening staan. Daarom is het tweede onderzoek nogmaals uitgevoerd met kleinere aantallen.



Ook hier is gemiddeld het stofverbruik lager dan het berekende stofverbruik. Dus ook bij kleinere aantallen is het gunstig om verschillende producten in een intekening te zetten. Het minimaal aantal producten van 5 is ook goed voor deze proef.

Tevens is er een soortgelijk onderzoek gedaan met verschillende maten van de bikini. Dit levert het volgende plaatje op:



Ook hier treedt een licht stofvoordeel op bij het mixen van maten.

### ***Eindconclusies nesteronderzoek***

Een minimum van vijf producten per intekening zorgt ervoor dat de stofbreedte efficiënt gebruikt wordt.

Wanneer verschillende producten en maten in een intekeningen worden gemixt, kost een product gemiddeld meestal minder stof dan wanneer intekeningen maar één product bevatten.

## Bijlage B: Snijlijsten van geteste werkorders

### Werkorder 4163

Behoefte WO

Type	Cup	38	40	42	44	46
601	B	26	32	28	8	4
601	C	20	40	40	28	8
801	B	12	14	12	6	4
801	C	14	18	20	14	1
991	B	78	100	98	58	22
991	C	52	100	112	94	48

Snijlijst systeem

Intekening	WO	Cup	38	40	42	44	46	lagen 601	lagen 801	lagen 991	meter
1	4163	B	2	2	2	0	0	13			2,67
2	4163	B	0	3	1	4	2	2			4,86
3	4163	C	2	4	4	2	0	10			5,66
4	4163	C	0	0	0	4	4	2			4,21
5	4163	B	2	2	2	1	0		6		3,18
6	4163	B	0	1	0	0	2		2		1,61
7	4163	C	3	4	5	3	0		4		7,04
8	4163	C	1	1	0	1	1		2		1,90
9	4163	B	2	2	2	1	0			30	3,18
10	4163	B	1	2	2	1	1			20	3,27
11	4163	B	0	2	1	5	2			2	4,86
12	4163	C	2	4	4	4	2			23	7,67
13	4163	C	2	2	6	0	0			3	4,57
14	4163	C	0	1	1	1	1			2	1,97

De eerste oplossing die door het JAVA-programma gegeven wordt is:

Snijlijst

Intekening	WO	Cup	38	40	42	44	46	lagen 601	lagen 801	lagen 991	meter
1	4163	B	2	2	2	0	0	13	5	12	7
1	4163	C	1	3	3	2	0	13	5	12	
2	4163	B	2	2	2	2	0	0	1	27	7,1
2	4163	C	1	2	2	2	0	0	1	27	
3	4163	B	0	2	2	0	2	1	0	10	6,73
3	4163	C	1	1	1	1	4	1	0	10	
4	4163	C	0	0	2	1	1	0	1	6	2,03
5	4163	B	0	2	0	4	2	2	0	1	6,77
5	4163	C	3	0	0	1	2	2	0	1	
6	4163	B	0	1	0	2	2	0	2	0	5,68
6	4163	C	4	1	1	1	0	0	2	0	

De snijlijst die door het JAVA-programma gegeven wordt voor kleuren apart:

Snijlijst

Intekening	WO	Cup	38	40	42	44	46	lagen 601	lagen 801	lagen 991	meter
1	4163	B	1	1	1	0	0	20			4,19
1	4163	C	1	2	2	1	0	20			
2	4163	B	2	3	2	2	1	4			6,68
2	4163	C	0	0	0	2	2	4			
3	4163	C	0	1	1	0	0			18	1,12
4	4163	B	1	1	1	0	0			12	2,34
4	4163	C	1	0	0	1	0			12	
5	4163	B	0	1	0	3	2			2	4,94
5	4163	C	1	0	1	1	1			2	
6	4163	B	3	3	3	2	0			26	6,82
6	4163	C	2	2	0	0	0			26	
7	4163	C	0	2	4	4	2			23	6,02
8	4163	B	0	1	1	0	1			20	1,93
8	4163	C	0	0	1	0	0			20	
9	4163	B	0	1	0	3	1			2	3,88
9	4163	C	0	1	0	1	1			2	

### Werkorders 5152 en 5158

Behoeft  
werkorders

wo	cup	36	38	40	42	44
5102	B	3	21	21	12	9
	C	2	18	20	13	9
	D	1	20	22	16	14
5158		18	46	59	33	10

De snijlijst van het systeem voor werkorder 5102 alleen:

Intekening	WO	Cup	36	38	40	42	44	lagen
1	5102	B	0	3	3	1	1	7
2	5102	B	2	0	0	2	1	2
3	5102	C	0	2	2	1	1	9
4	5102	C	1	0	1	2	0	2
5	5102	D	0	3	3	2	2	6
6	5102	D	0	1	2	2	1	2
7	5102	D	1	0	0	0	0	2

Met de hand:

Snijlijst

Intekening	WO	Cup	36	38	40	42	44	lagen	meter
1	5102	B		1	1			20	3,57
1	5102	C			1			20	
1	5102	D		1	1			20	
1	5158			2	3	1		20	
2	5102	B				1		10	2,61
2	5102	C		1		1		10	
2	5102	D				1	1	10	
2	5158		1			1	1	10	
3	5102	B	1				3	3	6
3	5102	C		2		1	3	3	
3	5102	D				2	1	3	
3	5158		3	2		1		3	
4	5102	B		1	1	2		1	3,28
4	5102	C	2	2				1	
4	5102	D	1		2		1	1	
4	5158							1	

Het JAVA-programma:

Snijlijst

Intekening	WO	Cup	36	38	40	42	44	lagen	meter
1	5102	B	0	1	1	0	0	18	4,16
1	5102	C	0	1	1	0	0	18	
1	5102	D	0	1	1	0	0	18	
1	5158		1	2	3	1	0	18	
2	5102	B	0	0	0	1	1	9	3,02
2	5102	C	0	0	0	1	1	9	
2	5102	D	0	0	0	1	1	9	
2	5158		0	1	0	1	1	9	
3	5102	B	1	1	1	1	0	4	6,26
3	5102	C	1	0	1	1	0	4	
3	5102	D	1	1	1	2	2	4	
3	5158		0	1	2	2	1	4	



Alternatief van het JAVA-programma:

Snijlijst

Intekening	WO	Cup	36	38	40	42	44	lagen	meter
1	5102	B	0	1	1	0	0	20	3,18
1	5102	C	0	0	1	0	0	20	
1	5102	D	0	1	1	0	0	20	
1	5158		0	2	2	1	0	20	
2	5102	B	0	0	0	1	1	9	4,59
2	5102	C	0	2	0	1	1	9	
2	5102	D	0	0	0	1	1	9	
2	5158		2	0	2	1	1	9	
3	5102	B	1	0	0	1	0	3	2,86
3	5102	C	0	0	0	1	0	3	
3	5102	D	0	0	0	2	1	3	
3	5158		0	2	0	1	0	3	
4	5102	B	0	1	1	0	0	1	4,28
4	5102	C	2	0	0	1	0	1	
4	5102	D	1	0	2	1	2	1	
4	5158		0	0	1	1	1	1	

5102 apart met JAVA-programma:

Intekening	WO	Cup	36	38	40	42	44	lagen	meter
1	5102	B	0	2	2	1	1	9	5,16
1	5102	C	0	2	2	1	1	9	
1	5102	D	0	2	2	1	1	9	
2	5102	B	1	1	1	1	0	4	3,95
2	5102	C	1	0	1	1	0	4	
2	5102	D	1	1	1	2	2	4	

Alternatief voor 5102 apart van het JAVA-programma:

Intekening	WO	Cup	36	38	40	42	44	lagen	meter
1	5102	B	0	2	2	1	1	9	5,16
1	5102	C	0	2	2	1	1	9	
1	5102	D	0	2	2	1	1	9	
2	5102	B	1	1	1	1	0	3	2,57
2	5102	C	0	0	0	1	0	3	
2	5102	D	0	0	1	2	1	3	
3	5102	C	2	0	2	1	0	1	3,77
3	5102	D	1	2	1	1	2	1	

5158 apart met JAVA-programma:

Intekening	WO	Cup	36	38	40	42	44	lagen	meter
1	5158		1	2	3	1	0	18	2,98
2	5158		0	2	1	3	2	5	4