NN31545.1934

ICW Nota 1934
december 1988

EEN KOSTENMODEL VOOR DE OPHOOGZANDVOORZIENING

Notaa van het Instituut zijn in principe interne communicatie-mededelingen, dus geen officiële publikaties.
Hun inhoud varieert sterk en kan zowel betrekking hebben op een eenvoudige weergave van cijferreeksen, als op een concluderende discussie van onderzoeksresultaten. In de meeste gevallen zullen de conclusies echter van voorlopige aard zijn omdat het onderzoek nog niet is afgesloten.
Genummerde notaa's komen niet voor verspreiding buiten het Instituut in aanmerking.

BIBLIOTHEEK
STARINGGEBOUW

Inhoud

Voorwoord iii

1. Inleiding

1.1 Algemeen 1
1.2 Doel van het onderzoek 1
1.3 Inhoud van het onderzoek 3

2. Schets van de ophoogzandvoorziening in Nederland

2.1 Inleiding 5
2.2 Behoefte aan ophoogzand 7
2.3 Voorziening in de ophoogzandbehoefte 7
2.4 Winning en transport 10
2.5 Vergunningen 13
2.6 Overheidsbeleid 15

3. Het kostenmodel

3.1 Inleiding 17
3.2 Uitgangspunten 18
3.3 De aankoopkosten van een terrein 18
3.4 De aanlegkosten van een zandwinput 22
3.5 De kosten van winning en transport bij transport per vrachtwagen 24
3.6 De kosten van winning en transport bij transport per pijpleiding 25
3.7 Het kostenmodel compleet 27

4. Keuze van de oplossingsmethoden

4.1 Inleiding 31
4.2 Alternatieve oplossingsalgorithmen 32
4.3 Het simplex algorithme voor transportproblemen 33
4.4 Het kostenmodel en het simplex algorithme voor transportproblemen 35

5. MPSX

5.1 Inleiding 43
5.2 Gebruikte programma's 44
5.3 Oplossen van een deelprobleem 45
5.4 Conclusies 45

6. ICW-oplossingsalgorithme en computerprogramma

6.1 Inleiding 47
6.2 Toepassen van het ICW-oplossingsalgorithme ter oplossing van het minimaliseringsprobleem 47
6.3 Van ICW-oplossingsalgorithme naar computerprogramma 52
6.4 Vereenvoudigen van de pijpleidingnetten en bepalen van de combinatiematrix, het hulpprogramma MODIFY 54
6.5 Besturen van het computerprogramma, het hoofdprogramma MAIN 56

CENTRALE LANDBOUWCATALOGUS
0000 0334 1225

11 APR. 1989

JSN 279268 *

6.6	Berekenen van de coëfficiënten van het kostenmodel, de subroutine RECALCO	57
6.7	Bepalen van een startoplossing voor de dummievariabelen, de subroutine ANSOL	57
6.8	Bepalen van de oplossing van het primale probleem, de subroutines PRISOL en RESIM	62
6.9	Transformeren van de oplossing van het primale probleem naar de oplossing van het duale probleem, de subroutines TRANS en CORRECT	71
6.10	Bepalen van de dummievariabelen, de subroutine ZEON	77
7.	Ontwikkelen en testen van het programma	
7.1	Inleiding	83
7.2	Test van de berekening van de spline functie variabelen	84
7.3	Test van de subroutine RECALCO	84
7.4	Test van de subroutine ANSOL	85
7.5	Test van de subroutine RESIM	85
7.6	Test van de subroutine PRISOL	88
7.7	Test van de berekening van de duale variabelen	89
7.8	Test van de subroutine TRANS	90
7.9	Test van de subroutine ZEON	91
7.10	Oorspronkelijke en laatste versie van het programma	91
7.11	Test van het hulpprogramma MODIFY	93
7.12	Twee runs van de laatste versie van het programma	93
8.	Het kostenmodel als beleidsondersteunend instrument	
8.1	Inleiding	95
8.2	Het tijdaspect ontbreekt	96
8.3	De grootte van de zandwinput wordt bepaald door de vraag	98
8.4	Er wordt geen specifieke korreldiameter gevraagd	99
8.5	Het zand wordt gewonnen middels een winzuiger	100
8.6	Het zand wordt getransporteerd per vrachtauto of per pijpleiding	101
8.7	Kostencategorieën	
8.7.1	De aankoopkosten van de benodigde terreinen	102
8.7.2	De aanlegkosten	106
8.7.3	De kosten van winning en transport bij transport per vrachtwagen	108
8.7.4	De kosten van winning en transport bij transport per pijpleiding	109
8.8	Organisatorische aspecten en kosten/baten	
8.8.1	Organisatorische aspecten	111
8.8.2	Kosten/baten	114
9.	Conclusies en aanbevelingen	117
	Literatuuroverzicht	119
	Bijlagen (separaat)	1 t/m 187

Voorwoord

Dit rapport vormt het verslag van het door mij verrichtte afstudeerwerk ter afronding van de studie Civiele Techniek aan de TUD (Technische Universiteit Delft). Het afstudeerwerk heeft bestaan uit een onderzoek naar een kostenmodel voor de ophoogzandvoorziening.

Het initiatief voor dit onderzoek is uitgegaan van het ICW (Instituut voor Cultuurtechniek en Waterhuishouding) te Wageningen, waar zowel het kostenmodel als een oplossingsmethode zijn ontwikkeld.

Het onderzoek is verricht bij de Vakgroepen POO. (Planning, Ontwerpen en Organisatie) en SSOR (Statistiek, Stochastiek en Operations Research), deel uitmakend van de Faculteit der Civiele Techniek respectievelijk de Faculteit der Technische Wiskunde en Informatica van de TUD.

Verantwoordelijke hoogleraren en afstudeerbegeleiders waren Prof.ir. H.P.S. van Lohuizen, ir. B. Polen en dr.ir. P. van der Veer (POO) en Prof.dr.ir. J.J.M. Evers en ir. Th.C.A. Mensch (SSOR). Vanuit het ICW is het onderzoek begeleid door drs. J. Vreke en ing. G.F.P. IJkelenstam.

Ik dank een ieder die mij geholpen heeft dit afstudeerwerk tot een goed einde te brengen. In het bijzonder dank ik mijn vrouw Wilma en mijn zonen Caspar en Boudewijn voor hun geduld en morele steun.

Delft, december 1988

Frank van Rest

1. Inleiding

1.1 Algemeen

Door de lage ligging, de slechte ontwatering en de geringe draagkracht van de bodem in grote delen van Nederland is er een ruime vraag naar ophoogzand.

Ophoogzand wordt gebruikt bij het bouwrijp maken van terreinen voor woningbouw en industrie, de aanleg van sportterreinen, wegen, spoorlijnen en dijken, voor cultuurtechnische verbetering van de bodem enz.

Het benodigde zand wordt voor het overgrote deel in Nederland zelf gewonnen; er wordt in zeer geringe mate ophoogzand ingevoerd doch dit blijft beperkt tot de grensstreken.

De winning van het zand, die grotendeels in particuliere handen is, is gebonden aan ontgrondingsvergunningen van de overheid. Deze vergunningen werden tot in het recente verleden verleend zonder dat hier een landelijk of een regionaal beleid aan ten grondslag lag. De winning van ophoogzand legt echter in aanzienlijke mate beslag op schaarse ruimte en beïnvloedt bestaande ecologische systemen. Vandaar dat de noodzaak is ingezien van een beleidsmatige aanpak van de vergunningverlening.

Het ICW heeft een kostenmodel ontwikkeld dat een bijdrage moet leveren aan de provinciale beleidsvorming t.a.v. de verlening van ontgrondingsvergunningen. Het model moet inzicht geven in de invloed van het aantal, de omvang en de ligging van de zandwinputten op de totale kosten van de ophoogzandvoorziening in een bepaald gebied. Dit om een verantwoorde keuze te kunnen maken tussen de diverse potentiële zandwinlocaties.

Uitgaande van het kostenmodel en een eveneens door het ICW ontwikkelde oplossingsmethode heeft het ICW een voorstel voor een afstudeeronderzoek geformuleerd. Dit voorstel, waarvan de formulering inclusief toelichting is opgenomen in bijlage I, is voorgelegd aan de Faculteit der Civiele Techniek. Het afstudeeronderzoek, waarvan dit rapport het verslag vormt, is mede op basis van dit voorstel tot stand gekomen.

In de navolgende twee paragrafen worden het doel en de inhoud van het afstudeeronderzoek besproken.

1.2 Doel van het onderzoek

Het doel van het onderzoek is geweest om vast te stellen in hoeverre het kostenmodel een bijdrage kan leveren aan de beleidsvorming t.a.v. de verlening van ontgrondingsvergunningen.

De bijdrage die het kostenmodel aan de beleidsvorming kan leveren, bestaat uit het geven van inzicht in de invloed van het aantal, de omvang en de ligging van de zandwinputten op de totale kosten van de ophoogzandwinning in een bepaald gebied. Een

bijkomend resultaat is dat met het kostenmodel de meest economische transportwijze wordt bepaald.

Zoals uit hoofdstuk 3 zal blijken, is het kostenmodel een minimaliseringsprobleem. Om te kunnen vaststellen of het kostenmodel toepassing heeft als beleidsondersteunend instrument, zal dit minimaliseringsprobleem moeten worden opgelost. Dit vereist een oplossingsmethode. Daarnaast moet het kostenmodel een zekere waarheidsgetrouwheid hebben, het moet praktisch bruikbaar zijn en de kosten die verbonden zijn aan het gebruik van het model, moeten opwegen tegen de baten.

Het onderzoek heeft zich daarom gericht op de volgende aspecten:

1. **De oplossingsmethoden waarmee het kostenmodel kan worden opgelost, in het bijzonder het algemene softwarepakket MPSX en de ICW-methode.**

2. **De grootte van de problemen die kunnen worden opgelost en de gevraagde rekentijd bij computerberekening.**

3. **De waarheidsgetrouwheid van het kostenmodel.**

4. **De praktische bruikbaarheid van het kostenmodel.**

5. **De kosten en baten die verbonden zijn aan het gebruik van het kostenmodel.**

Voor het onderzoek is uitgegaan van een proefgebied in de Betuwe ten oosten van Dodewaard (zie fig. 1). Van dit gebied zijn de voor het kostenmodel benodigde gegevens door het ICW verzameld. Deze gegevens zijn opgenomen in bijlage II.

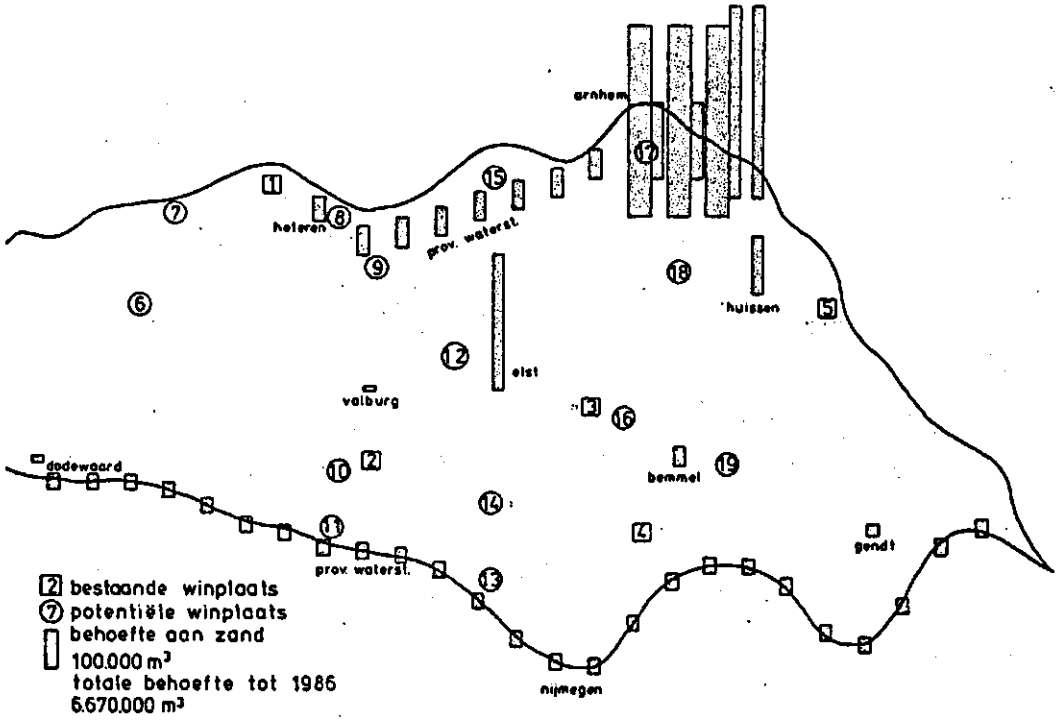


Fig. 1 Proefgebied met de ligging en omvang van de vraag en de ligging van potentiële winplaatsen ([7])

1.3 Inhoud van het onderzoek

Het afstudeeronderzoek heeft uit de volgende onderdelen bestaan:

Literatuurstudie - hoofstuk 2

Er is een beperkte literatuurstudie gedaan naar de ophoogzandwinning in Nederland. Gekeken is naar de gevraagde kwaliteit van het ophoogzand, naar vraag en aanbod van ophoogzand, naar de technieken van winning en transport alsmede naar de benodigde winningsvergunningen en het gevoerde overheidsbeleid t.a.v. de vergunningverlening.

ICW-kostenmodel - hoofstuk 3

Het ICW-kostenmodel, zoals beschreven in [7], [12] en [13], is bestudeerd en samengevat.

Keuze van de oplossingsmethoden - hoofdstuk 4

Er is een gemotiveerde keuze gemaakt voor twee oplossingsmethoden, het standaard softwarepakket MPSX en de ICW-methode. Daarnaast is aandacht besteed aan alternatieve oplossingsmethoden en is een richting aangegeven, die bij de ontwikkeling van een alternatieve methode gevolgd zou kunnen worden.

MPSX - hoofdstuk 5

Het minimaliseringsprobleem is opgelost met behulp van het standaard softwarepakket MPSX, waarbij gebruik is gemaakt van de invoergenerator GEN80 en het zelfgeschreven programma INPUT.

ICW-oplossingsmethode en computerprogramma - hoofdstukken 6 en 7

De ICW-oplossingsmethode, zoals geformuleerd in [12] en [13], is bestudeerd en samengevat. Voor deze methode is een computerprogramma ontwikkeld op basis van een door het ICW geschreven broncode. Zowel de uiteindelijke versie van het programma als de ontwikkeling en het testen zijn beschreven. De met het programma behaalde resultaten worden besproken.

Het kostenmodel als beleidsondersteunend instrument - hoofdstuk 8

Er is een analyse gemaakt van de bijdrage die het kostenmodel kan leveren aan de beleidsvorming t.a.v. het verlenen van ontgrondingsvergunningen. Hierbij is aandacht besteed aan de waarheidsgetrouwheid van het model, de mogelijkheden tot afweging van alternatieven, de mate van operationaliteit, de organisatorische aspecten en de kosten/baten.

Conclusies en aanbevelingen - hoofstuk 9

De in het afstudeeronderzoek getrokken conclusies zijn samengevat. Daarnaast zijn aanbevelingen gedaan voor verdere studie.

2. Schets van de ophoogzandvoorziening in Nederland

2.1 Inleiding

Dit hoofdstuk geeft de resultaten weer van een beperkte literatuurstudie naar de ophoogzandvoorziening in Nederland. Gekeken is naar de kwalitatieve en kwantitatieve aspecten van de ophoogzandvoorziening, naar de technieken van winning en transport alsmede naar de benodigde vergunningen en het overheidsbeleid t.a.v. de vergunningverlening.

Onder ophoogzand wordt in het algemeen verstaan zand met een korreldiameter van 63-210 μ ([15]). Ophoogzand wordt gebruikt bij het bouwrijp maken van terreinen voor o.a. woningbouw en industrie, de aanleg van sportterreinen, wegen, spoorlijnen en dijken, voor cultuurtechnische verbetering van de bodem enz. Het zand is hierbij nodig voor ophoging en om de draagkracht en ontwatering van de bodem te verbeteren.

Ophoogzand komt verspreid voor in vrijwel geheel Nederland evenals in de Waddenzee en de Noordzee; in fig. 2 is dit aangegeven.

Aan ophoogzand worden de volgende eisen gesteld:

- Redelijk doorlatend; de neerslag moet voldoende snel kunnen wegstromen.
- Voldoende draagvermogen; de pakking van de korrels dient voldoende te zijn.
- Beperkt verontreinigd; de meest voorkomende verontreinigingen zijn zout (zeezand), humus, diverse koolwaterstoffen en zware metalen (havenslib).

Wat de eisen t.a.v. doorlatendheid en draagvermogen concreet inhouden voor korrelverdeling, verdichtingsgraad en humusgehalte van het zand is voor wat betreft de wegenbouw opgenomen in "De eisen voor bouwstoffen in de wegenbouw 1978", opgesteld door Rijkswaterstaat en voor de aanleg van spoorlijnen in "De algemene bepalingen voor uitvoering van werken" van de Nederlandse Spoorwegen. Op de zoutbezwaren van zeezand wordt ingegaan in [15].

De aan ophoogzand te stellen eisen gelden in de ophoging, ze hoeven niet op de zandwinplaats te gelden:

- Indien de korrelverdeling van het zand op de zandwinplaats niet aan de eisen voldoet kan ze alsvolgt gewijzigd worden:
  - Het toepassen van ontslibbingsapparatuur.
  - De zeer fijne deeltjes op het stort afvoeren door het toepassen van hoge stroomsnelheden bij het opspuiten.
  - Eventuele stoorlagen (humus, klei) tijdens het winnen van het ophoogzand apart verwijderen.
- Een te hoog zoutgehalte van het zand kan alsvolgt verlaagd worden:
  - Het zand tijdens het transport per schip met zoet water spoelen en droogpompen.

. Het zand opslaan in depot, het zoute poriënwater zakt dan weg.
. Het zand opslaan in een zoetwaterbekken.
. Het toepassen van een ontziltingsinstallatie.

- De verdichtingsgraad van het zand op de winplaats is niet van belang voor de mechanische eigenschappen van het zand op het stort, bij de winning en het transport gaat de samenhang tussen de korrels toch grotendeels verloren.

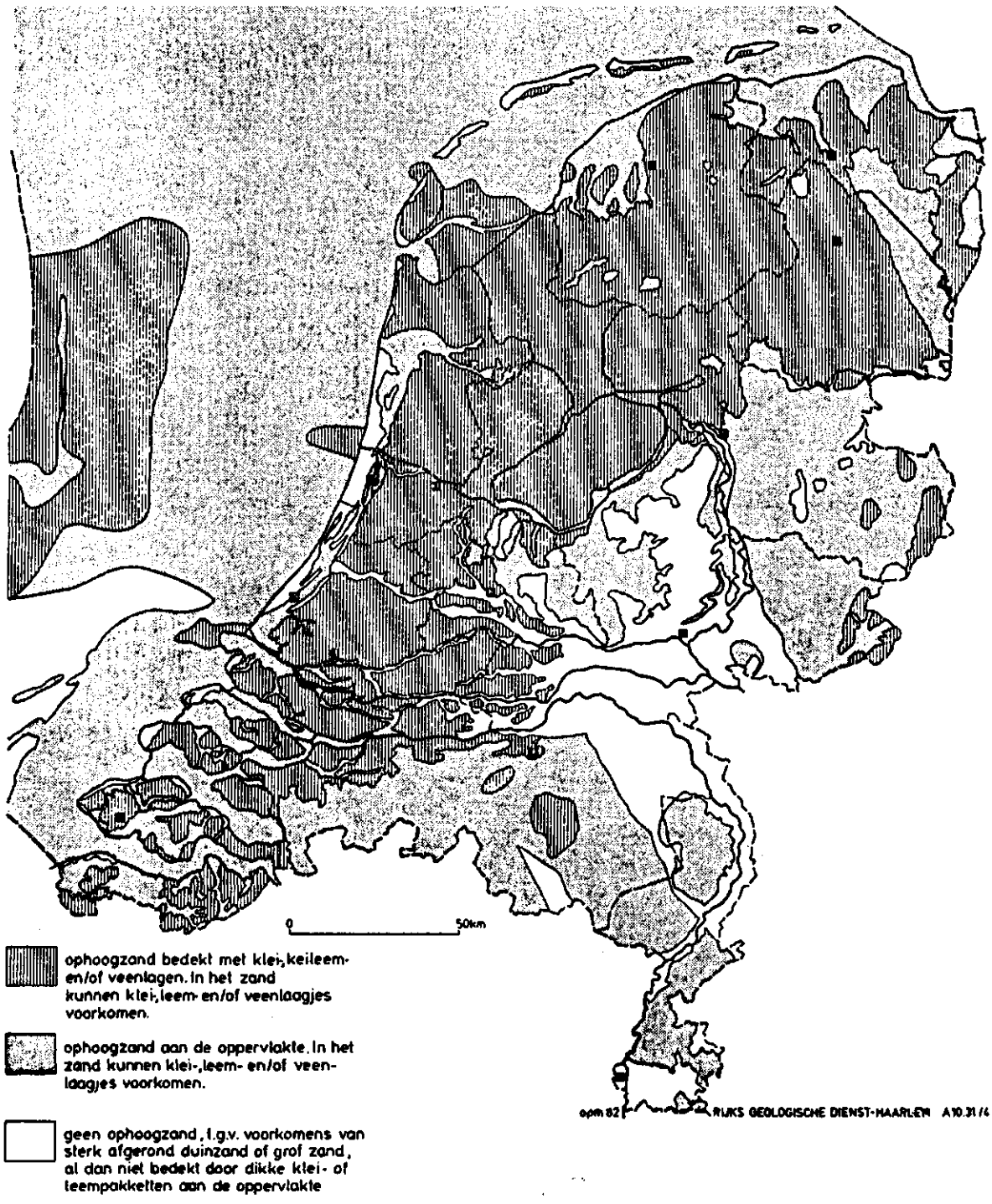


Fig. 2 Globale overzichtskaart van ophoogzandvoorkomens in Nederland aan of nabij de oppervlakte ([14])

2.2 Behoefte aan ophoogzand

In de achterliggende periode bedroeg de vraag naar ophoogzand in Nederland ca. 80 miljoen ton per jaar ([15]), hetgeen overeen komt met ca. 50 miljoen m^3 (1 ton zand = 0,6 m^3 dichtgepakt zand [5]). Door de provincies zijn ruwe schattingen gemaakt van de behoefte per provincie voor de periode 1984 t/m 1988. Deze schattingen zijn weergegeven in tabel 1.

Prov.	Gr.	Fr.	Dr.	Ov.	Ge.	Utr.	N.H.	Z.H.	Ze.	N.B.	Li.	Flev.	Tot.
min.	9	12	6	10	13	4	20	20	18	16	8	12	148
max.	14	16	8	18	20	6	25	30	21	28	15	13	214

Tabel 1 Minimale en maximale totale ophoogzandbehoefte per provincie voor de periode 1984 t/m 1988 in mln. m^3 ([15])

2.3 Voorziening in de ophoogzandbehoefte

In de vraag naar ophoogzand wordt vrijwel geheel door binnenlandse produktie voorzien, de in- en uitvoer van ophoogzand is zeer beperkt.

Hoe de produktie van ophoogzand in de periode 1980 t/m 1982 landelijk gespreid was is weergegeven in tabel 2.

Bij de winning van ophoogzand wordt onderscheid gemaakt tussen primaire en secundaire winning. Primaire winning heeft tot doel het winnen van ophoogzand. Bij secundaire winning is ophoogzand een bijprodukt, de winning van zand heeft dan een ander doel. Te denken valt hierbij aan het op diepte houden of verdiepen van vaargeulen, het verdiept aanleggen van wegen, de aanleg van recreatieprojecten of cultuurtechnische verbeteringen. De in hoeveelheden belangrijkste vorm van secundaire winning is het op diepte houden of verdiepen van vaargeulen, waarbij het meestal rijkswateren betreft. In tabel 3 is voor de periode 1980 t/m 1982 aangegeven hoe de verdeling totale winning - secundaire winning was van in rijkswateren gewonnen hoeveelheden ophoogzand.

Uit de tabellen 2 en 3 valt af te leiden dat de ophoogzandvoorziening in 1980, 1981 en 1982 voor resp. 49, 52 en 44% vanuit rijkswateren gewonnen is en dat het hier voor resp. 56, 41 en 51 % om secundaire winning ging.

In tabel 4 zijn meer recente produktiecijfers opgenomen, afkomstig van het CBS ([17]). Deze cijfers hebben betrekking op zelfgewonnen hoeveelheden ophoogzand. Uit een vergelijking met tabel 3 voor de jaren 1980 - 1982 blijkt dat de CBS-cijfers aanzienlijk lager liggen.

Gebied	Vergunning provincie			Vergunning MIN. V&W			Totaal		
	1980	1981	1982	1980	1981	1982	1980	1981	1982
Groningen	0,9	1,6	0,9	0,9	1,4	0,1	1,8	3,0	1,0
Friesland	2,2	1,1	0,5	3,5	2,8	1,2	5,7	3,9	1,7
Drenthe	1,4	1,6	1,3	0	0	0	1,4	1,6	1,3
Overijssel	1,8	1,2	1,2	0,2	0,2	0,2	2,0	1,4	1,4
Gelderland	3,0	3,0	3,0	0,2	0,1	0,2	3,2	3,1	3,2
Utrecht	2,0	2,0	2,0	0,2	0,2	0,2	2,2	2,2	2,2
N. Holland incl. IJmeer	1,4	0,5	0	4,1	3,3	1,8	5,5	3,8	1,8
Z. Holland	5,7	5,3	5,0	0,4	0,3	0,4	6,1	5,6	5,4
Zeeland	0,5	0	0	5,5	1,8	1,7	6,0	1,8	1,7
N. Brabant	2,2	3,7	2,9	1,0	1,2	1,2	3,2	4,9	4,1
Limburg	2,4	1,3	1,6	0,3	0,3	0,3	2,7	1,6	1,9
IJsselmeer en randmeren	-	-	-	1,3	4,2	4,2	1,3	4,2	4,2
Flevoland	-	-	-	2,4	5,0	1,2	2,4	5,0	1,2
Noordzee	-	-	-	2,9	2,4	1,5	2,9	2,4	1,5
Totaal	23,5	21,3	18,4	22,9	23,2	14,2	46,4	44,5	32,6

Tabel 2 Landelijke spreiding van de totaal gewonnen hoeveelheden ophoogzand, opgesplitst naar vergunningverlenende instantie, in mln. m^3 per jaar ([15])

Rijkswater	1980		1981		1982	
	totaal	waarvan secundair	totaal	waarvan secundair	totaal	waarvan secundair
Noordzee	2,9	2,2	2,4	2,2	1,5	1,4
Waddenzee incl. Eems/Dollard	6,1	0,7	5.3	0,7	2,0	0,7
IJmeer	1,4	0	2,2	0	1,1	0
IJsselmeer en randmeren	1,3	0,2	4,2	0,3	4,2	0,3
Zeeuwse wateren	5,5	5,5	1,8	1,8	1,7	1,7
Dir. Benedenrivieren RWS	0,7	0,7	0,7	0,7	0,7	0,7
Dir. Bovenrivieren RWS	0,5	0,5	0,7	0,7	0,6	0,6
Dir. Limburg RWS	1,2	1,2	1,1	1,1	1,3	1,3
Totaal	19,6	11,0	18,4	7,5	13,1	6,7

Tabel 3 Verdeling totale winning - secundaire winning van in rijkswateren gewonnen hoeveelheden ophoogzand, in mln. m^3 per jaar ([15])

De oorzaak hiervan is dat de LCCO-cijfers gebaseerd zijn op verleende ontgrondingsvergunningen terwijl de CBS-cijfers berusten op meldingen van bedrijven die zandwinning als hoofdactiviteit hebben. De produktie van de bedrijven, die het ophoogzand winnen voor specifieke projecten maar de zandwinning niet als hoofdactiviteit hebben (bijvoorbeeld grote aannemers) is niet opgenomen. Zoals uit de verschillen blijkt gaat het hier om grote hoeveelheden.

	1980	1981	1982	1983	1984	1985
Rivierzand	17760	11024	10202	11329	10600	10256
Groevezand	7971	8417	5561	5922	7329	9673
Duin- en zeezand	7243	7547	5709	11187	10238	9727
Totaal	32975	27014	21472	28438	28167	29657
W.v. gewonnen in:						
Gr., Fr. en Dr.	2761	4061	1566	1269	1936	2450
Ov. en Gld.	3252	3174	4343	9391	4468	4549
Utr., N.H. en Z.H.	19272	15712	12291	11625	14599	16380
Zeel. en N.Br.	6740	3584	2742	5589	6396	5074
Li.	948	482	530	564	767	1204

Tabel 4 In Nederland gewonnen hoeveelheden ophoogzand 1980 t/m 1985, in mln. kg ([17])

Bij de uitvoering van grotere werken wordt het ophoogzand rechtstreeks vanuit de winput in het werk gebracht. Er is hier geen sprake van tussenhandel. Naar schatting bedraagt dit gedeelte van de ophoogzandvoorziening naar hoeveelheid ca. driekwart van het totaal ([15]). Bij de uitvoering van kleinere werken is er wel sprake van tussenhandel, de zogenaamde zandwinkels. Deze veelal wat kleinere bedrijven leveren het zand rechtstreeks vanuit de winput of vanuit depot.

Als vervanging voor ophoogzand kunnen bulkafvalstoffen worden toegepast. De bulkafvalstoffen die hier het meest voor in aanmerking komen zijn bouw- en sloopafval en baggerspecie. Andere bulkafvalstoffen zoals vliegas, bodemas, diverse slakken, fosforzuurgips en mijnsteen zijn of geschikt voor hoogwaardiger toepassingen dan ophogen of ongeschikt vanwege gebrek aan stabiliteit of de aanwezigheid van uitloogbare verontreinigingen. Door het beperkte aanbod en de bezwaren voor het milieu (hetgeen met name geldt voor verontreinigde baggerspecie) speelt de toepassing van bulkafvalstoffen ter vervanging van ophoogzand geen belangrijke rol ([15]).

Een andere mogelijkheid om in de behoefte aan ophoogzand te voorzien is hergebruik. Niet meer in gebruik zijnde terreinophogingen, weg- en spoorweglichamen, dijken, defensiewerken en

industrie- en haventerreinen kunnen worden afgegraven en het gewonnen zand kan opnieuw voor ophogingsdoeleinden worden gebruikt. Het is niet bekend om welke hoeveelheden het hier gaat.

Een recente ontwikkeling is het toepassen van schuimbeton voor ophogingsdoeleinden. Het betreft hier partiële ophogingen (bijvoorbeeld voor de aanleg van wegen of sportvelden) op zeer slappe grond. Het aantal projecten waarbij deze methode is toegepast, is echter nog zeer beperkt.

2.4 Winning en transport

De meest gebruikelijke vorm van winning van ophoogzand is de natte winning, waarbij het zand onder water gewonnen wordt. De winwerktuigen die hierbij over het algemeen worden toegepast zijn in zoet water de profielzuiger (fig. 3) en de cutterzuiger (fig. 4) en op zee de sleephopperzuiger (fig. 5). In bijzondere gevallen (bijvoorbeeld bij het baggeren in talud) worden ook wel de baggermolen (fig. 6), de grijperkraan (fig. 7) en de natte graafschop (fig. 8) toegepast.

Wanneer bij aanvang van de winning geen waterplas aanwezig is, dient een put gegraven te worden alvorens het natte winwerktuig kan worden ingezet. Dit gebeurt meestal met een dragline of een grijperkraan.

Droge winning komt in Nederland veel minder vaak voor. Door de hoge grondwaterstand blijft deze vorm van winning veelal beperkt tot oppervlakteontgrondingen, die vrijwel alleen worden toegepast in agrarische gebieden. Meestal is er dan bovendien sprake van secundaire winning, het primaire doel is dan een cultuurtechnische verbetering. De bij droge winning toegepaste winwerktuigen zijn de dragline, de grijperkraan en incidenteel de bottomscraper.

Doordat ophoogzand over geheel Nederland verspreid voorkomt, wordt het ophoogzand doorgaans op een betrekkelijk korte afstand van de plaats van gebruik gewonnen. Het transport van ophoogzand gaat dus veelal over korte afstanden.

De transportwijzen die het meest worden toegepast zijn transport per pijpleiding, varend transport bestaande uit zelfvarend transport (fig. 9) of baktransport en transport over de weg. De transportband en het spoor worden in Nederland weinig gebruikt als transportmiddel voor ophoogzand (het spoor wordt wel door de Nederlandse Spoorwegen gebruikt bij transport van ophoogzand voor eigen gebruik).

Pijpleidingtransport wordt toegepast bij grote hoeveelheden; de initiële kosten voor de aanleg van de zuig-persinstallatie met bijbehorende leidingen zijn hoog.

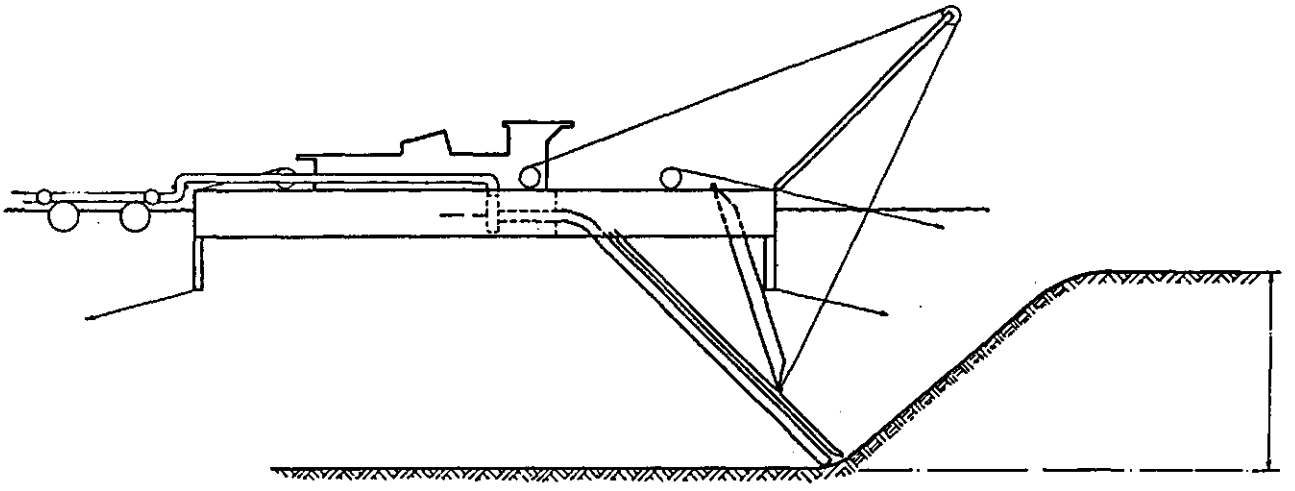


Fig. 3 Profielzuiger

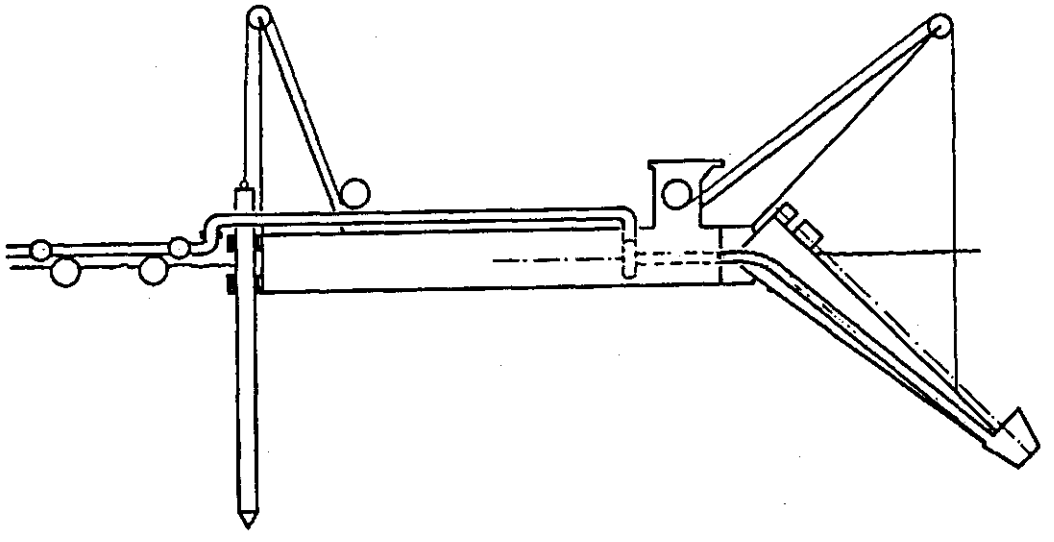


Fig. 4 Cutterzuiger

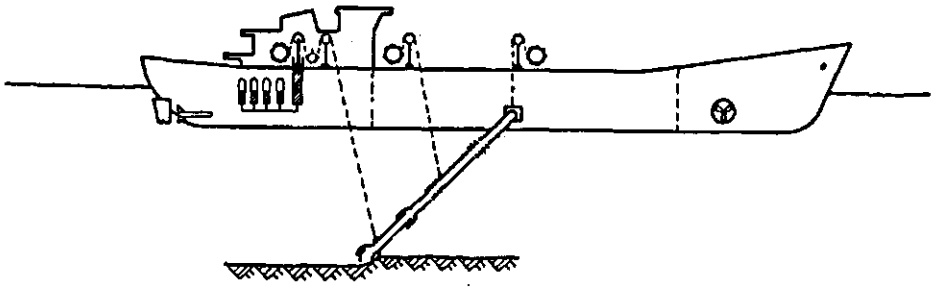


Fig. 5 Sleephopperzuiger

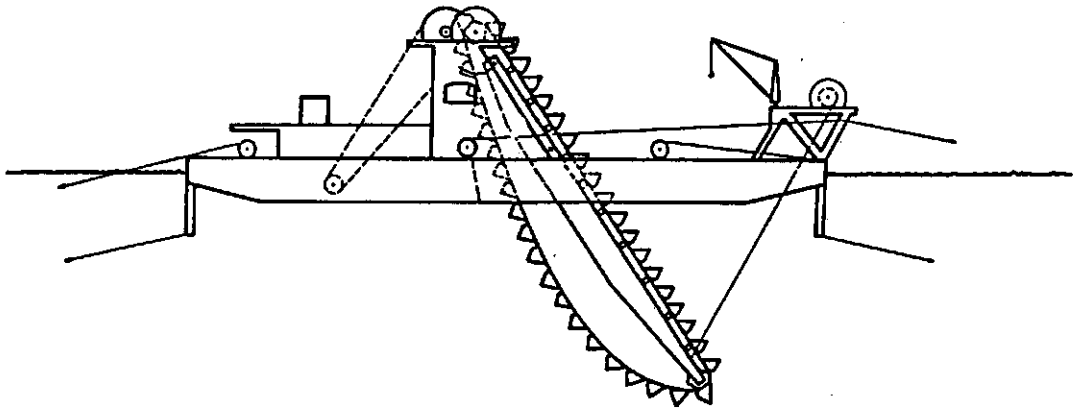


Fig. 6 Baggermolen

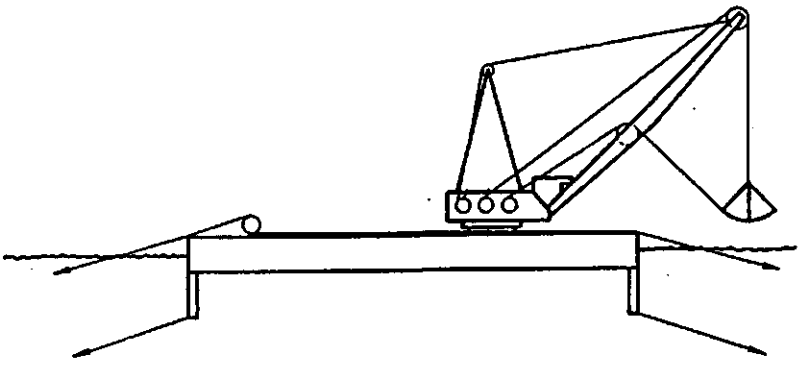


Fig. 7 Grijperkraan

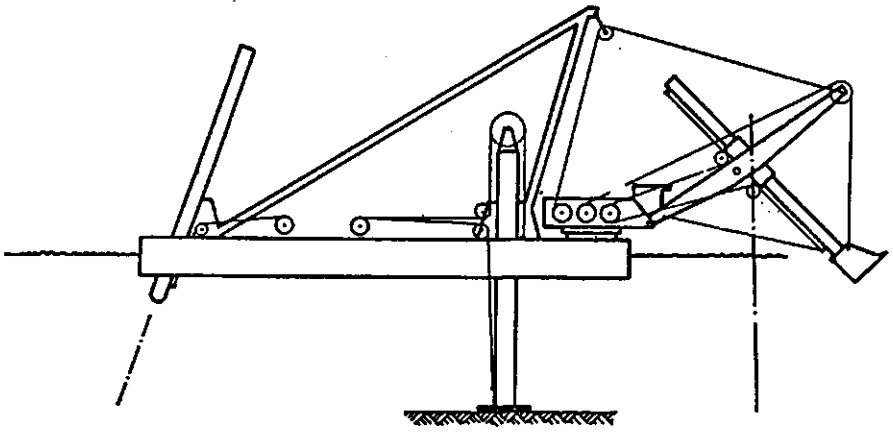


Fig. 8 Natte graafschop

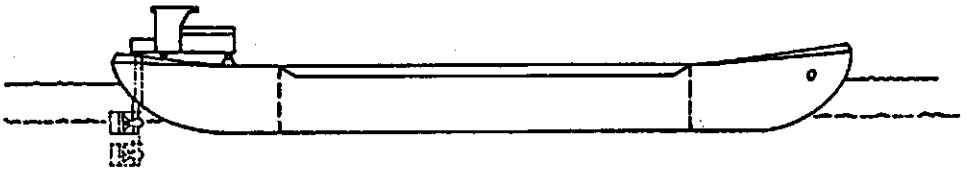


Fig. 9 Zelfvarende transportbak

Varend transport wordt vooral toegepast over grotere afstanden. Het vereist een goede vaarinfrastructuur waardoor in het westen van het land verhoudingsgewijs meer varend transport plaatsvindt dan in het oosten.

Transport over de weg wordt vooral toegepast over korte afstanden. Het transport is kwetsbaar, doordat regen en hoge grondwaterstanden de berijdbaarheid van de terreinen kan aantasten. Eventueel benodigde bouwwegen kunnen extra kosten met zich mee brengen. Daarnaast heeft transport over de weg een negatieve invloed op de verkeersveiligheid. Tegenover deze nadelen staat dat vrijwel elke plaats bereikbaar is.

2.5 Vergunningen

Voor de meeste ontgrondingen is volgens de Ontgrondingenwet van 1971 een ontgrondingsvergunning vereist. De winning van ophoogzand is vrijwel altijd aan een vergunning gebonden.

De bevoegdheid tot het verlenen van ontgrondingsvergunningen ligt bij:

- De Minister van Verkeer en Waterstaat voor de zee, de rijkswateren inclusief het zomerbed van de rijksrivieren en stromen, voor zover de Rivierenwet hierop niet van toepassing is en voor de ingepolderde gedeelten van het IJsselmeer, die nog niet provinciaal zijn ingedeeld.
- Gedeputeerde Staten van de provincie, waarin het te ontgronden perceel gelegen is in de overige gevallen, zij het dat vergunningverlening in het winterbed van rijksrivieren en stromen in overeenstemming met de Minister van Verkeer en Waterstaat dient te geschieden.

De Ontgrondingenwet is een zogenaamde kaderwet, die de verdere uitwerking in verordeningen overlaat aan de vergunningverlenende instanties. Bij Algemene Maatregel van Bestuur (Rijksoverheid) c.q. Provinciale Verordening hebben deze instanties regels gesteld betreffende de ontgrondingen. Deze regels hebben betrekking op:

- Het werkplan volgens welke de ontgrondingen zullen worden uitgevoerd.
- De toestand waarin de percelen na ontgronding moeten worden gebracht c.q. de verplichting om in plaats daarvan andere percelen in een bepaalde toestand te brengen of een som geld af te dragen.
- De zekerheden die gegeven moeten worden omtrent de betaling van de kosten van uitvoering en onderhoud van de voorgeschreven werken (bankgaranties e.d.).

Alvorens een vergunningverlenende instantie een beslissing neemt over een aanvraag, dient hij alle bij de aanvraag betrokken instellingen te horen en belanghebbenden de gelegenheid te geven hun eventuele bezwaren kenbaar te maken (zie het procedureschema in fig. 10).

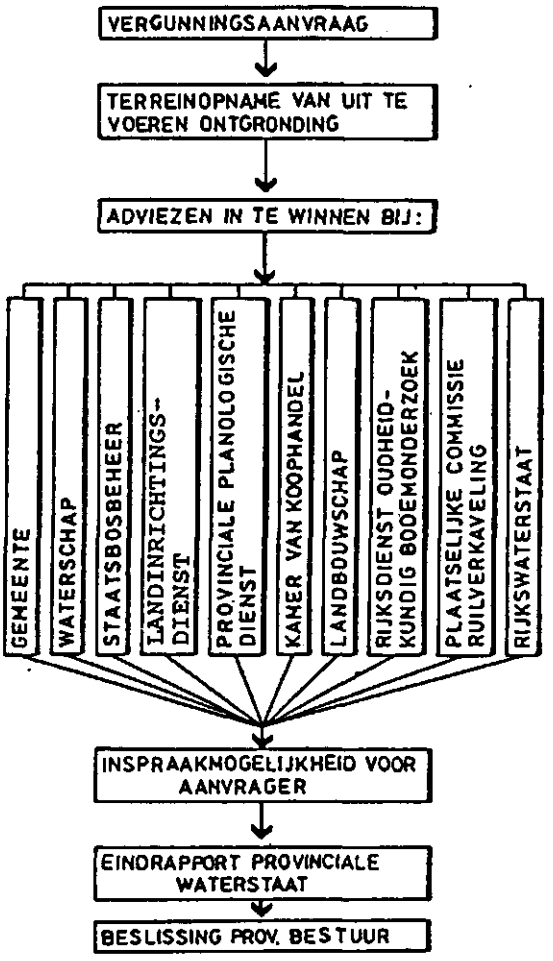


Fig. 10 Te doorlopen procedure bij het verlenen van een ontgrondingsvergunning ([5])

Bij de beslissing dienen alle bij de ontgronding betrokken belangen te worden afgewogen. Deze belangen zijn van tweeërlei aard:

Gediende belangen

- Belangen van de ontgronder die baten wenst te trekken uit de verkoop van de te winnen specie.
- Belangen van de eigenaar die streeft naar baten uit de ontgronding en eventuele verbetering van zijn eigendom.
- Belangen van de werken waarvoor de specie moet dienen.
- Belangen die betrekking hebben op het terrein dat achterblijft na de ontgronding.

Te schenden belangen

- Belangen die voortvloeien uit het gebruik en de bestemming van de ontgronding.

- Belangen van het aan het terrein gegeven planologische bestemming die niet gerealiseerd kan worden bij het uitvoeren van de ontgronding.
- Natuurwetenschappelijke en landschappelijke belangen.
- Belangen van eigenaars en gebruikers van belendende percelen.

De afstemming tussen de Ontgrondingenwet en de Wet op de Ruimtelijke Ordening is niet optimaal. Zo kan het voorkomen dat, nadat een ontgrondingsvergunning is verleend in het kader van de Ontgrondingenwet, planologische regelingen in het kader van de Wet op de Ruimtelijke Ordening (bestemmingsplanaanpassing, aanlegvergunning) het in produktie nemen van een zandwinning tegenhouden of vertragen. Dit probleem doet zich met name voor bij grote ontzandingen.
De afstemmingproblematiek Ontgrondingenwet - Wet op de Ruimtelijke Ordening is een belangrijke reden voor de (in voorbereiding zijnde) herziening van de Ontgrondingenwet.

2.6 Overheidsbeleid

Het overheidsbeleid t.a.v. de ophoogzandwinning maakt deel uit van het totale ontgrondingenbeleid. Dit op basis van de Ontgrondingenwet gevoerde beleid kan alsvolgt gekenschetst worden:

- De overheid komt formeel pas in actie wanneer een ontgrondingsvergunning wordt aangevraagd.
- Het beleid is sterk gedecentraliseerd; er zijn twaalf vergunningverlenende instanties nl. de rijksoverheid en de elf provincies.
- Het is een ad hoc beleid; elke aanvraag voor een ontgrondingsvergunning wordt op zich behandeld en beoordeeld.
- Het is een beleid waarin de samenhang met het beleid t.a.v. de ruimtelijke ordening vaak onvoldoende gewaarborgd is.

Toch heeft het ontgrondingenbeleid zich in de afgelopen jaren in een meer planmatige richting ontwikkeld. Alle vergunningverlenende instanties uitgezonderd de provincie Zeeland, die zich voor haar zandwinning op rijkswateren richt, hebben het te voeren beleid vastgelegd in ontgrondingsplannen c.q. nota's en/of streekplannen. Het beleid is vastgelegd in richtlijnen die per instantie kunnen verschillen. In het algemeen hebben de richtlijnen betrekking op:

- Situering van de winplaatsen.
- Oppervlakteontgrondingen.
- Bestemming en inrichting van de winplaatsen na ontzanding.
- Eigendom en exploitatie van de winplaatsen tijdens en na de ontzanding.
- Technische voorwaarden zoals minimale afstanden van de winplaatsen tot wegen en bebouwing, minimale breedte van de niet te vergraven oevers, maximale windiepte, afwerking van de oevers enz.

Voor de concrete inhoud van de richtlijnen van de verschillende provincies en de Minister van Verkeer en Waterstaat wordt verwezen naar [15] en de provinciale plannen c.q. rijksnota's.

Het geschetste beleid is zoals gezegd sterk gedecentraliseerd. De centrale overheid verleent wel advies bij het formuleren van de provinciale richtlijnen (o.a. in de nota Landelijke Gebieden) doch coördineert de activiteiten van de vergunningverlenende instanties niet. Onderling overleg over de aanwijzing van nieuwe winplaatsen en een systematische afweging tussen zandwinning uit landlocaties of rijkswateren vindt dan ook niet plaats.

De hierboven genoemde bezwaren die aan het huidige ontgrondingenbeleid verbonden zijn zullen ondervangen worden:

- De samenhang tussen het ontgrondingenbeleid en de ruimtelijke ordening zal verbeteren bij het in werking treden van de herziene Ontgrondingenwet.
- Voor de korte termijn (1984 tot 1989) is in [15] een beleid geformuleerd voor de ophoogzandvoorziening. Hierin wordt onder andere aandacht besteed aan de coördinatie van de activiteiten van verschillende van de vergunningverlenende instanties.
- Voor de lange termijn wordt momenteel een beleid voor de oppervlaktedelfstoffenvoorziening ontwikkeld; [14] is hier een voorloper van.

3. Het kostenmodel

3.1 Inleiding

In dit hoofdstuk worden de bij het formuleren van het kostenmodel gehanteerde uitgangspunten beschreven, wordt de bepaling van de verschillende kostencategorieën behandeld en wordt het kostenmodel in de vorm van vergelijkingen weergegeven. Er worden echter geen formules afgeleid of theoretische achtergronden behandeld. Hiervoor wordt verwezen naar de volgende nota's:

- ICW-nota 1335 ([12]); in deze nota is het kostenmodel geformuleerd.
- ICW-nota 1168 ([7]); aan deze nota zijn de formules ontleend waarmee de vaste en variabele winnings- en transportkosten worden berekend.
- ICW-nota 1343 ([13]); in deze nota is de benadering van de relatie tussen oppervlakte en inhoud van een zandwinput door een lineaire spline functie beschreven.

De functie van het kostenmodel is het uit het oogpunt van kostenminimalisering maken van de optimale keuze uit een aantal potentiële zandwinobjecten bij een gegeven vraag naar zand (gevraagde hoeveelheden per vraagpunt bekend). Dit betekent dat voor elk vraagpunt wordt bepaald vanuit welke zandwinputten welke hoeveelheden zand verkregen worden en op welke wijze deze getransporteerd worden, zodanig dat voor ieder vraagpunt aan de vraag voldaan wordt en de totale kosten verbonden aan de winning en het transport van het zand minimaal zijn. Hiermee is dan bekend welke zandwinobjecten in produktie worden genomen en welke transportwijzen worden toegepast.

Het model is opgebouwd uit een te minimaliseren doelstellingsfunctie en een aantal beperkingen, zoals dit in de Operations Research gebruikelijk is. De doelstellingsfunctie bestaat uit de som van alle in het model opgenomen kosten. Hierbij wordt onderscheid gemaakt tussen vaste en variabele kosten. De vaste kosten zijn onafhankelijk van de hoeveelheid geproduceerd of getransporteerd zand, terwijl de variabele kosten hiermee evenredig zijn.

In paragraaf 3.2 worden de uitgangspunten, die bij het formuleren van het model gehanteerd zijn, beschreven. In de paragrafen 3.3 t/m 3.6 wordt de bepaling van de verschillende kostencategorieën voor één zandwinput beschreven en worden de bijbehorende beperkingen geformuleerd. In paragraaf 3.7 wordt het volledige kostenmodel beschreven. De doelstellingsfunctie bestaat hierbij uit de som van de kosten van de verschillende zandwinputten en het stelsel beperkingen bestaat uit de som van de beperkingen per zandwinput, waaraan een aantal noodzakelijke beperkingen is toegevoegd.

3.2 Uitgangspunten

De belangrijkste bij het formuleren van het kostenmodel gehanteerde uitgangspunten zijn:

- Het tijdaspect blijft in het model buiten beschouwing; de kostenminimalisering wordt uitgevoerd over een bepaalde periode, waarbinnen de vraag naar zand per vraagpunt geraamd wordt.
- De grootte van een zandwinput wordt bepaald door de vraag, met dien verstande dat er per zandwinobject een minimale vraag is, waarbeneden het object niet in produktie wordt genomen.
- Bij de vraag naar zand wordt niet gespecificeerd welke korreldiameter gevraagd wordt; de korreldiameter van het zand heeft wel invloed op de winnings- en transportkosten.
- De winning van het zand vindt plaats middels een drijvende winzuiger.
- Het transport van het zand vindt plaats per vrachtauto of per pijpleiding.
- De volgende kostencategorieën zijn in het model opgenomen:
  . De aankoopkosten van de benodigde terreinen.
  . De aanlegkosten; dit zijn de kosten die gemaakt worden voordat een zandwinput in produktie genomen kan worden.
  . De kosten van de winning en het transport van het zand.

In hoofdstuk 8 wordt ingegaan op de invloed die deze uitgangspunten hebben op de betekenis van het kostenmodel als beleidsondersteunend instrument.

3.3 De aankoopkosten van een terrein

De aankoopkosten van een terrein zijn evenredig met de oppervlakte van het terrein:

$$CAK = pg.R(q) \qquad (3.1)$$

met CAK : aankoopkosten van een terrein [f]
pg : grondprijs [f/m^2]
R(q): terreinoppervlakte als functie van de putproduktie [m^2]
q : putproduktie [m^3]

In het kostenmodel wordt een zandwinput gekarakteriseerd door de putproduktie (de hoeveelheid zand die uit betreffende winput gewonnen wordt). Daarom moet er om de aankoopkosten van het voor een zandwinput benodigde terrein te bepalen een relatie worden gelegd tussen de produktie (de inhoud) van de winput en de oppervlakte van het benodigde terrein. Voor een zandwinput met een omgekeerde piramidevorm (fig. 11) geldt de volgende relatie:

$$R(q) = \{2t.v + (6t.q)^{1/3}\}^2;\ rq(0) \le q \le rq(1) \qquad (3.2)$$

$$rq(1) = \tfrac{4}{3}t^2d^3$$

met d : dikte van de laag winbaar zand [m]
rq(0): minimaal te winnen hoeveelheid zand in geval de winput in produktie genomen wordt [m^3]
rq(1): hoeveelheid te winnen zand waarbij de winput overgaat van omgekeerde piramidevorm in afgeknotte omgekeerde piramidevorm [m^3]
t : talud van de winput [-]
v : dikte van de niet uit zand bestaande bovenlaag [m]

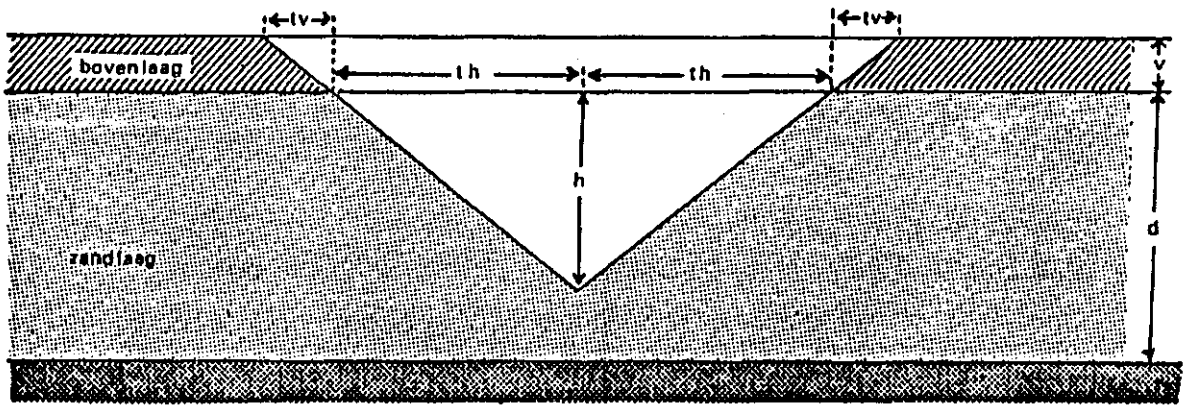


Fig. 11 Doorsnede van een winput met een omgekeerde piramidevorm

Voor een winput met een afgeknotte omgekeerde piramidevorm (fig. 12) geldt de volgende relatie:

$$R(q) = \{(\frac{q}{d} - \frac{1}{3}t^2d^2)^{1/2} + t.d + 2t.v\}^2;\ q \geq rq(1) \qquad (3.3)$$

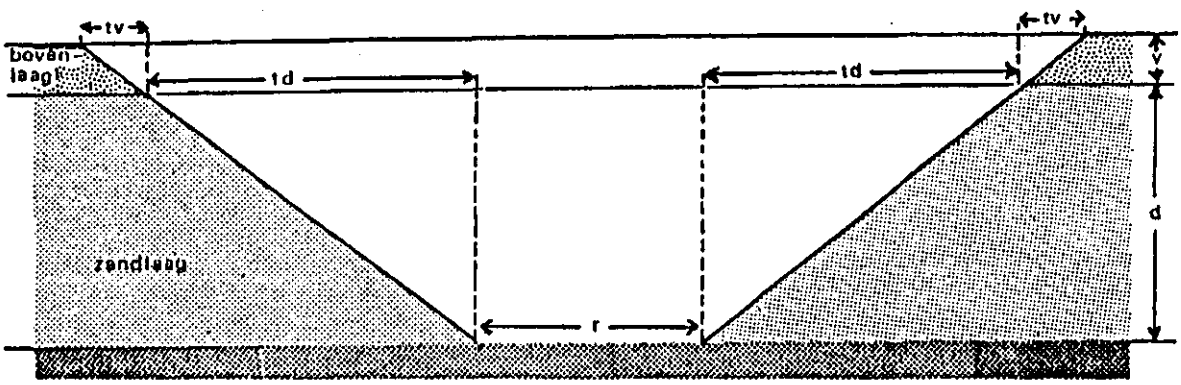


Fig. 12 Doorsnede van een winput met een afgeknotte omgekeerde piramidevorm

Het kostenmodel is geformuleerd als een lineair programmeringsmodel, waarin alleen lineaire functies mogen voorkomen. De relaties (3.2) en (3.3) moeten daarom gelineariseerd worden. Hierbij wordt gebruik gemaakt van een lineaire spline functie, een benadering van niet-lineaire functies door een aantal lijnstukken. In fig. 13 is deze benadering weergegeven.

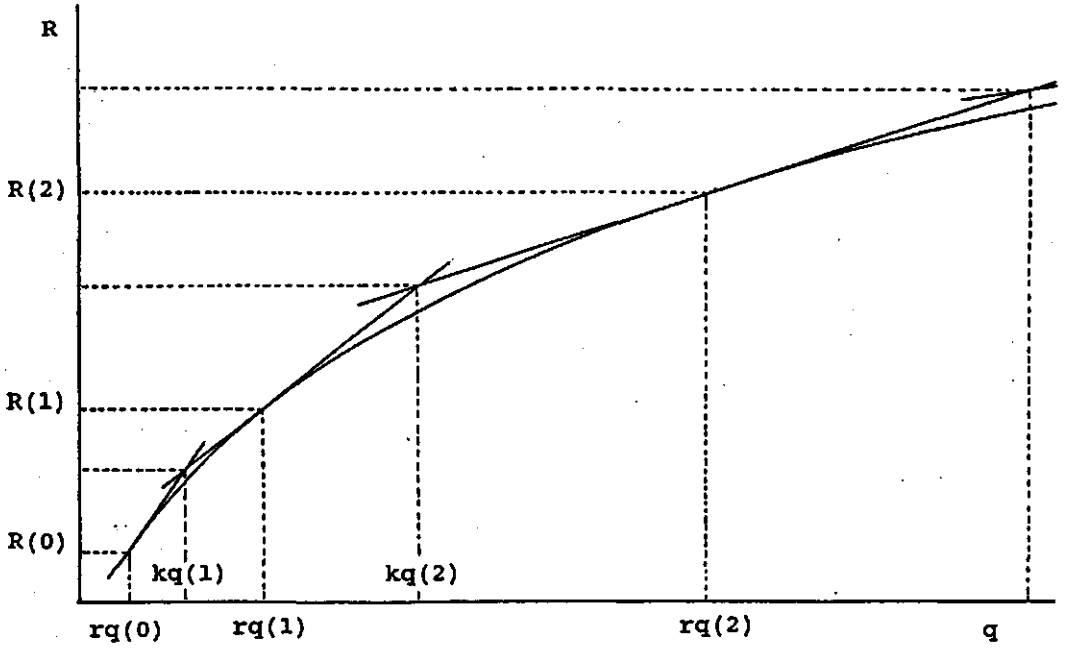


Fig. 13 Benadering van een niet-lineaire functie door een lineaire spline functie

De lineaire spline functie als benadering van R(q) kan alsvolgt geformuleerd worden:

$$RS(q) = R(rq(0)) + R'(rq(0))qx(0) + \sum_{k=1}^{K} \{gam(k)qx(k)\} \qquad (3.4)$$

$$qx(0) = \max\{0,\ q-rq(0)\} \qquad (3.5)$$

$$qx(k) = \max\{0,\ q-kq(k)\};\ k = 1,\ldots,\ K \qquad (3.6)$$

$$gam(k) = R'(rq(k)) - R'(rq(k-1));\ k = 1,\ldots,\ K \qquad (3.7)$$

$$rq(k) = \frac{\frac{1}{3}t^2 d^3 + d\{\frac{kq(k)}{d} - \frac{1}{3}t^2 d^2\}^2}{\frac{rq(k-1)}{d} - \frac{1}{3}t^2 d^2};\ k = 1,\ldots,\ K \qquad (3.8)$$

$$kq(1) = \frac{R(rq(1)) - R(rq(0)) + rq(0)R'(rq(0)) - rq(1)R'(rq(1))}{R'(rq(0)) - R'(rq(1))} \tag{3.9}$$

$$kq(k) = \tfrac{1}{3}t^2d^3 + \tag{3.10}$$

$$+ d\left[\frac{\{R(rq(k-1))(dR'(rq(k-1)) - 1)\rho\}^{1/2} + (1+\rho)(t.d + 2tv)^2}{dR'(rq(k-1)) - 1 - \rho}\right]$$

$$k = 2, \ldots, K$$

$$R'(q) = 4t(6t.q)^{-2/3}\{2t.v + (6t.q)^{1/3}\};\ rq(o) \le q \le rq(1) \tag{3.11}$$

$$R'(q) = \frac{1}{d}\left[1 + \frac{t.d + 2t.v}{\{\frac{q}{d} - \frac{1}{3}t^2d^2\}^{1/2}}\right];\ q \ge rq(1) \tag{3.12}$$

met K : aantal spline functie intervallen [-]
kq(k): zandproduktie waarbij het k-de knikpunt in de spline functie optreedt [m^3]
qx(k): hoeveelheid zand die méér geproduceerd wordt dan de hoeveelheid kq(k) [m^3]
ρ : relatieve afwijking $\frac{RS(q)-R(q)}{R(q)} < \rho$ [-]
rq(k): hoeveelheid zand waarbij de benadering RS(rq(k)) raakt aan de functie R(rq(k)) in het k-de raakpunt [m^3]
RS(q): lineaire benadering van de functie R(q) [m^2]
R'(q): $\frac{dR(q)}{dq}$ [m^{-1}]

Bij deze benadering is uitgegaan van een maximaal toegestane relatieve afwijking ρ. In [12] en [13] is ook een spline functie benadering geformuleerd, uitgaande van een maximaal toegestane absolute afwijking δ. Hierbij wordt het nauwkeurigheidscriterium $RS(q) - R(q) \le \delta$ gehanteerd. In het kostenmodel is de benadering met het relatieve nauwkeurigheidscriterium opgenomen, hetgeen een arbitraire keuze is.

De aankoopkosten van het terrein voor één winput worden alsvolgt in het kostenmodel opgenomen:

$$CAK = pg[\{R(rq(0)) - rq(0)R'(rq(0))\}d1 + R'(rq(0))q + \sum_{k=1}^{K} \{gam(k)qx(k)\}] \quad (3.13)$$

$$ST^{*}\ q - QA.d1 \leq 0 \quad (3.14)$$

$$q - qx(k) - kq(k)d4(k) \geq 0;\ k = 1,\ldots, K \quad (3.15)$$

$$qx(k) - Q(k)d4(k) \leq 0;\ k = 1,\ldots, K \quad (3.16)$$

$$q,\ qx(k) \geq 0$$

$$d1,\ d4(k):\ 0\text{-}1 \text{ variabelen}$$

met d1 : dummievariabele (0-1) die aangeeft of de winput in produktie genomen wordt [-]
d4(k): dummievariabele die aangeeft of de putproduktie groter is dan kq(k) [-]
Q(k) : QA-kq(k) [m^3]
QA : totale vraag naar zand [m^3]

De aankoopkosten van een terrein zijn afhankelijk van de putproduktie; het betreft hier dus variabele kosten.

3.4 De aanlegkosten van een zandwinput

In het kostenmodel is er van uit gegaan dat het zand gewonnen wordt met een drijvende winzuiger die het zand via een drijvende pijpleiding naar de wal verperst. Vandaar wordt het zand via een landleiding naar de vraagpunten verperst of in depot gebracht. In het laatste geval wordt het zand dan vanuit het depot m.b.v. een hydraulische kraan in vrachtwagens geladen en naar de vraagpunten getransporteerd.

Onder de aanlegkosten worden de volgende kosten verstaan:

- De kosten van het graven van de put, nodig om de drijvende winzuiger te kunnen inzetten.
- De kosten van aanleg en opruimen van de winzuiger.
- De kosten van aanleg en opruimen van de drijvende pijpleiding.
- De kosten van aanleg en opruimen van de pijpleiding die van de winput naar het depot loopt (bij transport per vrachtwagen).
- De kosten van aanleg en opruimen van het deel van de pijpleiding dat zich op de winlocatie bevindt (bij transport per pijpleiding).

Het betreft hier vaste kosten die onafhankelijk zijn van de hoeveelheid zand die uit de winput gewonnen wordt.

---

* ST (Subject To) betekent "onder de voorwaarde dat".

De aanlegkosten van een zandwinput worden alsvolgt in het kostenmodel opgenomen:

$$CAL = CA.d1 + CB.d1 \qquad (3.17)$$

$$ST \quad q - QA.d1 \leq 0 \qquad (3.18)$$

$$q \geq 0$$

$$d1: 0\text{-}1 \text{ variabele}$$

met CA : vaste aanlegkosten van de zandwinput [f]
CB : vaste winningskosten bij transport per pijpleiding [f]
CAL: aanlegkosten van de zandwinput [f]; deze zijn nul wanneer de put niet in produktie wordt genomen (d1 = 0).
d1 : dummievariabele die aangeeft of de winput in produktie genomen wordt [-]
q : putproduktie [m^3]
QA : totale vraag naar zand [m^3]

De aanlegkosten van een zandwinput bij transport per vrachtwagen verschillen van die bij transport per pijpleiding. Dit komt omdat bij transport per vrachtwagen tot de vaste aanlegkosten de kosten van aanleg en opruimen van de pijpleiding die naar het zanddepot loopt gerekend worden en bij transport per pijpleiding de kosten van aanleg en opruimen van het deel van de pijpleiding dat zich op de winlocatie bevindt. Hierdoor kunnen deze kosten in de doelstellingsfunctie niet zonder meer gekoppeld worden aan de dummievariabele d1. Dit is opgelost door in het kostenmodel een dummievariabele d2 op te nemen die aangeeft of er vanuit een winput al dan niet transport per vrachtwagen plaatsvindt. De aanlegkosten van een zandwinput worden nu gedefinieerd als de vaste aanlegkosten plus de vaste winningskosten bij transport per pijpleiding. De aanlegkosten van een zandwinput bij transport per vrachtwagen worden gedefinieerd als de vaste aanlegkosten plus het verschil tussen de vaste winningskosten bij transport per vrachtwagen en bij transport per pijpleiding. Aan dit verschil wordt in de doelstellingsfunctie de dummievariabele d2 gekoppeld.

Bij het winklaar maken van een zandwinput dient de niet uit zand bestaande bovenlaag te worden verwijderd. Deze bovenlaag is vaak bruikbaar in de keramische industrie of voor de ophoging van dijken. Tegenover de kosten verbonden aan het verwijderen van de bovenlaag staan de opbrengsten uit de verkoop of het gebruik van deze laag. Bij de formulering van het kostenmodel is er van uitgegaan dat het verschil tussen deze kosten en opbrengsten opgenomen is in de grondprijs.

3.5 De kosten van winning en transport bij transport per vrachtwagen

Bij de winning en het transport bij transport per vrachtwagen worden de volgende kosten onderscheiden:

- De vaste winningskosten; deze zijn gedefinieerd als het in paragraaf 3.4 genoemde verschil tussen de vaste winningskosten bij transport per vrachtwagen en bij transport per pijpleiding.
- De variabele winningskosten; dit zijn de kosten van de huur van het materieel, de brandstofkosten en de personeelskosten. Deze kosten zijn evenredig met de gewonnen hoeveelheid zand en afhankelijk van de gemiddelde korreldiameter van het zand (groffer zand vereist een hoger pompvermogen). Omdat de invloed van de korreldiameter op de totale kosten in dit geval relatief gering is (de korreldiameter heeft bij transport per vrachtwagen alleen invloed op de variabele winningskosten en niet op de transportkosten) wordt ze verwaarloosd.
- De variabele transportkosten; dit zijn eveneens de kosten van de huur van het materieel, de brandstofkosten en de personeelskosten. Deze kosten zijn evenredig met de hoeveelheid te transporteren zand en met de transportafstand.

De kosten van winning en transport bij transport per vrachtwagen worden alsvolgt in het kostenmodel opgenomen:

$$CTW = (CW - CB).d2 + \sum_{j=1}^{m} \{qw(j)pw(j)\} \qquad (3.19)$$

$$ST \quad \sum_{j=1}^{m} \{qw(j)\} - QA.d2 \leq 0 \qquad (3.20)$$

$$qw(j) \geq 0$$

d2: 0-1 variabele

met CTW : kosten van winning en transport bij transport per vrachtwagen [f]
CB : vaste winningskosten bij transport per pijpleiding [f]
CW : vaste winningskosten bij transport per vrachtwagen [f]
d2 : dummievariabele die aangeeft of er vanuit de winput transport per vrachtwagen plaatsvindt [-]
m : totaal aantal vraagpunten [-]
pw(j): prijs van het transport per vrachtwagen [f/m^3]
QA : totale vraag naar zand [m^3]
qw(j): hoeveelheid zand die per vrachtwagen wordt getransporteerd van de winput naar het vraagpunt j [m^3]

In [7] is de prijs van het transport per vrachtwagen alsvolgt gedefinieerd:

$$pw(j) = 2,6 + 0,35dw(j) \qquad (3.21)$$

met dw(j): gewogen transportafstand van de winput naar het vraagpunt j [km]

De factor 2,6 geeft de variabele winningskosten en de uitdrukking 0,35dw(j) de variabele transportkosten. De gemiddelde rijsnelheden van de vrachtwagens bedragen op de verharde weg 40 km/u, op de onverharde weg 20 km/u en in het terrein 10 km/u. Omdat de transportprijs samenhangt met de rijtijd van de vrachtwagens en dus met de rijsnelheden dienen deze snelheden in de transportprijs verwerkt te worden. Dit gebeurt door gewogen transportafstanden te hanteren. De werkelijke transportafstanden worden hierbij vermenigvuldigd met de factoren 4, 2 en 1 voor respectievelijk rijden in het terrein, op de onverharde weg en op de verharde weg.

3.6 De kosten van winning en transport bij transport per pijpleiding

Bij de winning en transport van zand bij transport per pijpleiding worden de volgende kosten onderscheiden:

- De vaste transportkosten; dit zijn de kosten van aanleg en afvoeren van het leidingennet inclusief de tussenstations. Deze kosten zijn evenredig met de lengte van de leidingen en afhankelijk van de gemiddelde korreldiameter van het zand.
- De vaste winningskosten; opgenomen in de aanlegkosten van de winput (paragraaf 3.4).
- De variabele winningskosten; dit zijn de kosten van huur van het materieel, de brandstofkosten en de personeelskosten. Ze zijn evenredig met de gewonnen hoeveelheid zand. De invloed van de gemiddelde korreldiameter van het zand op deze kosten wordt evenals bij het transport per vrachtwagen verwaarloosd.
- De variabele transportkosten; dit zijn de kosten van huur van het materieel, de brandstofkosten en de personeelskosten. Deze kosten zijn evenredig met de hoeveelheid zand die getransporteerd wordt en met de afgelegde weg in het leidingennet, waardoor de kosten per m^3 per vraagpunt verschillen. Tevens zijn deze kosten afhankelijk van de gemiddelde korreldiameter van het zand.

Doordat bij het transport vanuit een winput naar de verschillende vraagpunten mogelijkerwijs voor een deel dezelfde trajecten worden gebruikt (samenvallende trajecten) kunnen de vaste kosten van het leidingennet niet zonder meer aan de vraagpunten worden gekoppeld.

Daarom wordt de volgende procedure gevolgd:

- Het leidingennet dat alle vraagpunten met de winput verbindt wordt gedefinieerd; in fig. 14 is het leidingennet van het proefgebied weergegeven.
- De trajecten worden genummerd van 1 t/m S, ongeacht of het samenvallende of niet samenvallende trajecten betreft.

- De coëfficiënten coef(s,j) worden bepaald; coef(s,j) = 1 wanneer traject s deel uitmaakt van het kortste leidingtracé dat de winput met het vraagpunt j verbindt; anders geldt coef(s,j) = 0.

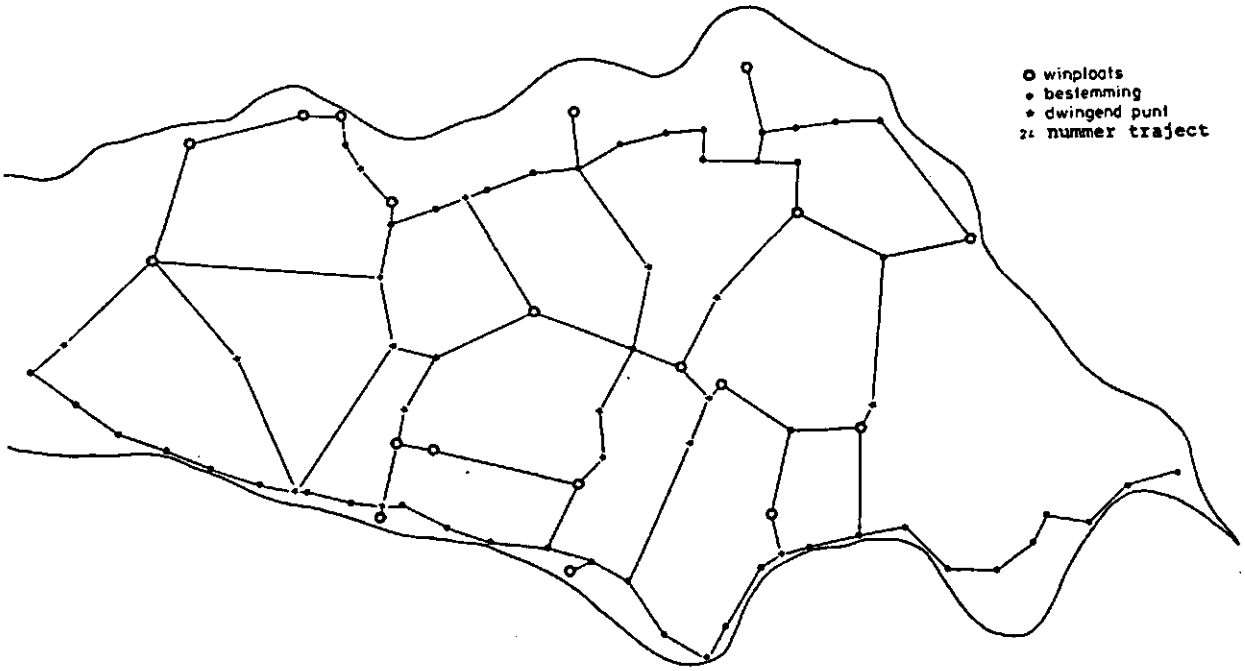


Fig. 14 Leidingennet van het proefgebied

De kosten van winning en transport bij transport per pijpleiding worden alsvolgt in het kostenmodel opgenomen:

$$CTB = \sum_{s=1}^{S} \{CT(s)d3(s)\} + \sum_{j=1}^{m} \{qb(j)pb(j)\} \qquad (3.22)$$

$$ST \quad \sum_{j=1}^{m} \{coef(s,j)qb(j)\} - QB(s)d3(s) \leq 0 \qquad (3.23)$$

$$QB(s) = \sum_{j=1}^{m} \{coef(s,j)a(j)\} \qquad (3.24)$$

$$qb(j) \geq 0$$

d3(s): 0-1 variabele

met a(j) : vraag naar zand van vraagpunt j [m^3]
coef(s,j): 0-1 coëfficiënt die aangeeft of traject s deel uitmaakt van het leidingtracé dat de winput met vraagpunt j verbindt [-]
CTB : kosten van winning en transport bij transport per pijpleiding [f]

CT(s) : vaste kosten van traject s [f]
d3(s) : dummievariabele die aangeeft of traject s gebruikt (en dus aangelegd) wordt [-]
m : totaal aantal vraagpunten [-]
pb(j) : prijs van het winnen van het zand en het transporteren naar vraagpunt j [f/m^3]
qb(j) : hoeveelheid zand die per pijpleiding van de winput naar het vraagpunt j getransporteerd wordt [m^3]
QB(s) : maximale hoeveelheid zand die via traject s getransporteerd wordt [m^3]
S : totaal aantal trajecten [-]

In [7] worden de vaste transportkosten verbonden aan traject s alsvolgt gedefinieerd:

$$CT(s) = \alpha.D(s) \tag{3.25}$$

met α : kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand [f/km]
D(s): lengte van leidingtraject s [km]

De prijs van de winning en het transport van het zand per pijpleiding is volgens [7] alsvolgt gedefinieerd:

$$pb(j) = 0{,}66 + \delta.DB(j) \tag{3.26}$$

met $DB(j)$: $\sum_{s=1}^{S} \{coef(s,j)D(s)\}$, de lengte van het leidingtracé dat de winput met vraagpunt j verbindt [km]
δ : kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand [f/km]

De factor 0,66 geeft de variabele winningskosten en de uitdrukking $\delta.DB(j)$ de variabele transportkosten.

3.7 Het kostenmodel compleet

In de paragrafen 3.3 t/m 3.6 zijn de diverse kosten voor één zandwinput geformuleerd. De totale kosten voor één zandwinput zijn:

$$TC = CAK + CAL + CTW + CTB \tag{3.27}$$

De volgende kostenparameters worden gedefinieerd:

$$A = pg\{R(rq(0) - rq(0)R'(rq(0))\} + CA + CB$$

$$B = CW - CB$$

$$E(s) = CT(s)$$

$$F(j) = pg.R'(rq(0)) + pw(j)$$

$$G(j) = pg.R'(rq(0)) + pb(j)$$

$$H(k) = pg.gam(k)$$

De doelstellingsfunctie van het kostenmodel voor één zandwinput wordt geformuleerd door de vergelijkingen (3.13), (3.17), (3.19) en (3.22) in vergelijking (3.27) te substitueren onder gebruikmaking van de hierboven gedefinieerde parameters. De beperkingen van het kostenmodel bestaan uit de beperkingen (3.14) t/m (3.16), (3.18) welke identiek is aan (3.14)), (3.20) en (3.23). Het kostenmodel kan nu alsvolgt geschreven worden:

$$TC = A.d1 + B.d2 + \sum_{s=1}^{S} \{E(s)d3(s)\} + \sum_{j=1}^{m} \{F(j)qw(j)\} \qquad (3.28)$$

$$+ \sum_{j=1}^{m} \{G(j)qb(j)\} + \sum_{k=1}^{K} \{H(k)qx(k)\}$$

$$ST \quad \sum_{j=1}^{m} \{qw(j) + qb(j)\} - QA.d1 \leq 0 \qquad (3.14)$$

$$\sum_{j=1}^{m} \{qw(j)\} - QA.d2 \leq 0 \qquad (3.20)$$

$$\sum_{j=1}^{m} \{coef(s,j)qb(j)\} - QB(s)d3(s) \leq 0; \; s = 1,\ldots, S \qquad (3.23)$$

$$\sum_{j=1}^{m} \{qw(j) + qb(j)\} - qx(k) - kq(k)d4(k) \geq 0; \qquad (3.15)$$

$$k = 1,\ldots, K$$

$$qx(k) - Q(k).d4(k) \leq 0; \; k = 1,\ldots, K \qquad (3.16)$$

$$qw(j), \; qb(j) \text{ en } qx(k) \geq 0$$

d1 ,d2 ,d3(s) en d4(k): 0-1 variabelen

De totale kosten voor alle zandwinputten zijn:

$$TCG = \sum_{i=1}^{n} \{TC(i)\} \qquad (3.29)$$

met n : totaal aantal zandwinputten [-]
TC(i): totale kosten voor zandwinput i [f]
TCG : totale kosten van de zandvoorziening voor het gehele proefgebied [f]

In het vervolg van dit verslag wordt de index i gebruikt om een winput aan te geven.

Aan het kostenmodel dienen de volgende beperkingen te worden toegevoegd:

$$\sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} = a(j);\ j = 1,\ldots, m \qquad (3.30)$$

Voor iedere vraagpunt dient aan de vraag voldaan te worden.

$$\sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} \geq rq(i,0)d1(i);\ j = 1,\ldots, m \qquad (3.31)$$

Indien zandput i in produktie wordt genomen dient minimaal de hoeveelheid rq(i,0) gewonnen te worden.

Bij het minimaliseren wordt de beperking (3.30) vervangen door:

$$\sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} \geq a(j);\ j = 1,\ldots, m \qquad (3.32)$$

Na oplossing van het probleem moet gecontroleerd worden of aan de gelijkheid (3.30) is voldaan; zo niet dan moet de beperking (3.33) aan het kostenmodel worden toegevoegd en moet er opnieuw geminimaliseerd worden.

$$\sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} \leq a(j) \quad \text{voor } j = 1, m \qquad (3.33)$$

Het kostenmodel geschreven als minimaliseringsprobleem kan nu alsvolgt geformuleerd worden:

$$\begin{aligned} \min TCG = & \sum_{i=1}^{n} \{A(i)d1(i)\} + \sum_{i=1}^{n} \{B(i)d2(i)\} + \sum_{i=1}^{n} \{ \sum_{s=1}^{S} \{E(i,s)d3(i,s)\}\} + \\ & + \sum_{i=1}^{n} \{ \sum_{j=1}^{m} \{F(i,j)qw(i,j)\}\} + \sum_{i=1}^{n} \{ \sum_{j=1}^{m} \{G(i,j)qb(i,j)\}\} + \\ & + \sum_{i=1}^{n} \{ \sum_{k=1}^{K} \{H(i,k)qx(i,k)\}\} \end{aligned} \qquad (3.34)$$

$$\text{ST} \quad \sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} \geq a(j);\ j = 1,\ldots, m \qquad (3.35)$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - rq(i,0)d1(i) \geq 0; \qquad (3.36)$$
$$i = 1,\ldots, n$$

$$-\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} + QA.d1(i) \geq 0; \qquad (3.37)$$
$$i = 1,\ldots, n$$

$$-\sum_{j=1}^{m} \{qw(i,j)\} + QA.d2(i) \geq 0;\ i = 1,\ldots, n \qquad (3.38)$$

$$-\sum_{j=1}^{m} \{coef(i,s,j)qb(i,j)\} + QB(i,s)d3(i,s) \geq 0; \qquad (3.39)$$
$$i = 1,\ldots, n$$
$$s = 1,\ldots, S$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - qx(i,k) - kq(i,k)d4(i,k) \geq 0;$$
$$i = 1,\ldots, n$$
$$k = 1,\ldots, K$$
$$(3.40)$$

$$-qx(i,k) + Q(i,k)d4(i,k) \geq 0;\ i = 1,\ldots, n \qquad (3.41)$$
$$k = 1,\ldots, K$$

$qw(i,j)$, $qb(i,j)$ en $qx(i,k) \geq 0$

$d1(i)$, $d2(i)$, $d3(i,s)$ en $d4(i,k)$: 0-1 variabelen

4. Keuze van de oplossingsmethoden

4.1 Inleiding

Het in het voorgaande hoofdstuk besproken kostenmodel is een programmeringsprobleem dat de vorm heeft van een gemengd geheeltallig lineair minimaliseringsprobleem. Voor het gebruik van het kostenmodel als beleidsondersteunend instrument moet dit minimaliseringsprobleem voor verschillende situaties m.a.w. met verschillende modelparameters worden opgelost.

Het oplossen van dit probleem zal volgens een bepaalde methode moeten verlopen; er zal gebruik gemaakt moeten worden van een oplossingsalgorithme*.

Er kunnen twee typen oplossingsalgorithmen onderscheiden worden:

1. Algemene algorithmen; dit zijn bestaande algorithmen die een grote verscheidenheid aan lineaire programmeringsproblemen kunnen oplossen. Voor deze algorithmen zijn over het algemeen standaard softwarepakketten beschikbaar.

2. Specifieke algorithmen; dit zijn algorithmen die voor een specifiek probleem of een specifiek soort problemen zijn ontwikkeld. De computerprogramma's die voor dit type algorithmen beschikbaar zijn, zijn over het algemeen niet standaard doch speciaal ontwikkeld door instellingen of bedrijven die zich met het oplossen van dergelijke problemen bezig houden.

In het afstudeeronderzoek is gebruik gemaakt van beide typen algorithmen en programma's:

- MPSX (Mathematical Programming System eXtended, [19]); een standaard softwarepakket voor lineaire programmering van IBM waarmee onder gebruikmaking van de MIP-optie (Mixed Integer Programming, [20]) gemengd geheeltallige lineaire programmeringproblemen kunnen worden opgelost.
- Het ICW-algorithme; een algorithme dat door het ICW specifiek voor het betreffende minimaliseringsprobleem ontwikkeld is ([12] en [13]). Het computerprogramma voor dit algorithme is in het kader van het onderzoek ontwikkeld.

De keuze voor beide algorithmen is eenvoudig te verklaren:

- Met MPSX, een pakket dat tot begin 1987 ter beschikking stond van de gebruikers van de Centrale Digitale Installatie van het Rekencentrum van de TUD, kunnen gemengd geheeltallige lineaire programmeringsproblemen worden opgelost. Omdat het kostenmodel als een dergelijk probleem geformuleerd is en het afstudeer-

---

* "Algorithme" wordt hier gebruikt in de betekenis van "oplossingsprocedure"; veelal wordt hiermee een iteratieve oplossingsprocedure bedoeld.

onderzoek binnen de TUD verricht is, lag de keuze voor MPSX voor de hand.
- Het voorstel voor een afstudeeronderzoek, zoals dit door het ICW aan de Faculteit der Civiele Techniek is voorgelegd (zie bijlage I), richt zich in hoofdzaak op het ICW-algorithme. Hoewel betreffend voorstel niet zonder meer is overgenomen, heeft het ICW-algorithme een belangrijke plaats behouden in het onderzoek.

De keuze voor beide algorithmen is dus uit praktische overwegingen gemaakt en de beschikbare tijd liet geen uitgebreid onderzoek naar andere algorithmen toe. Toch kan men zich afvragen of er geen ander algorithme is waarmee het probleem op een efficiënte wijze kan worden opgelost. Deze vraag is des te actueler omdat het afstudeeronderzoek geen praktisch toepasbaar oplossingsalgorithme heeft opgeleverd (zie paragraaf 7.12 en hoofdstuk 9).

In dit hoofdstuk wordt daarom aandacht besteed aan een alternatief oplossingsalgorithme. In paragraaf 4.2 wordt aangegeven op welk alternatief algorithme de aandacht zich gericht heeft en op welke gronden. In paragraaf 4.3 wordt nader ingegaan op een onderdeel van dit alternatieve algorithme nl. het simplex algorithme voor transportproblemen. In paragraaf 4.4 wordt onderzocht of het simplex algorithme voor transportproblemen in combinatie met de branch-and-bound techniek geschikt is voor het oplossen van het minimaliseringsprobleem. Dit blijkt alleen het geval te zijn indien de formulering van het probleem wordt aangepast. Daarom wordt een aantal maatregelen voorgesteld waardoor het minimaliseringsprobleem met het simplex algorithme voor transportproblemen in combinatie met de branch-and-bound techniek kan worden opgelost.

Het gebruik van MPSX en het ICW-algorithme worden besproken in de hoofdstukken 5 en 6.

4.2 Alternatieve oplossingsalgorithmen

Hoewel MPSX zeker niet het enige standaard softwarepakket is, waarmee gemengd geheeltallige lineaire programmeringsproblemen kunnen worden opgelost, is niet getracht het kostenmodel met andere standaardpakketten op te lossen. De reden hiervoor is dat van de andere pakketten geen prestaties verwacht worden, die afwijken van die van MPSX*. Dit wordt onderbouwd door ervaringen die binnen het ICW met andere standaardpakketten zijn opgedaan.

Ook is vooraf geen aandacht besteed aan alternatieve algorithmen, gebaseerd op het ICW-algorithme. Zoals uit de hoofdstukken 6 en 7 zal blijken, is het ICW-algorithme tijdens het ontwikkelen van het computerprogramma wel aangepast.

---

* Zoals uit hoofdstuk 5 zal blijken, is het niet mogelijk gebleken om het kostenmodel voor het volledige proefgebied m.b.v. MPSX op te lossen.

Bij het onderzoek naar mogelijke alternatieve oplossingsalgorithmen heeft de aandacht zich gericht op algorithmen die gebaseerd zijn op een algemene oplossingsmethode, maar waarbij gebruik wordt gemaakt van de specifieke kenmerken van het kostenmodel.

Het kostenmodel is geformuleerd als een gemengd geheeltallig lineair programmeringsprobleem. Er is sprake van continue en van discrete (0-1) variabelen en van een lineaire doelstellingsfunctie en lineaire beperkingen. De algemene oplossingsmethode voor dit soort problemen is het standaard simplex algorithme* in combinatie met de branch-and-bound techniek. MPSX is een voorbeeld van een oplossingsalgorithme dat gebaseerd is op deze methode. Voor de theoretische achtergronden van het standaard simplex algorithme en van de branch-and-bound techniek wordt verwezen naar bestaande literatuur (o.a. [2] en [4]). In bijlage III wordt de branch-and-bound techniek, toegepast op een probleem met continue en discrete variabelen, nader toegelicht.

Deze algemene oplossingsmethode brengt met zich mee dat voor een gemengd geheeltallig lineair probleem van enige omvang een groot aantal lineaire programmeringsproblemen met het standaard simplex algorithme moet worden opgelost. Omdat de problemen ieder een zekere omvang hebben en er bovendien een groot aantal problemen moet worden opgelost, leggen de computerprogramma's die op deze methode gebaseerd zijn in hoge mate beslag op computergeheugen en rekentijd. Zeer grote problemen kunnen soms zelfs helemaal niet worden opgelost (zie de voetnoot op de voorgaande bladzijde).

Nu zijn er type problemen waarvoor efficiëntere oplossingsalgorithmen bestaan dan het standaard simplex algorithme. Eén daarvan is het transportprobleem. Omdat het kostenmodel het karakter heeft van een transportprobleem, er is immers sprake van aanbodpunten, van vraagpunten en van transport, heeft de aandacht zich gericht op het oplossingsalgorithme voor transportproblemen.

4.3 Het simplex algorithme voor transportproblemen

Voor transportproblemen bestaat er een speciaal oplossingsalgorithme, het simplex algorithme voor transportproblemen. Met dit algorithme kunnen transportproblemen op een efficiënte wijze worden opgelost.

Een transportprobleem kan alsvolgt geformuleerd worden:

$$\min \sum_{i=1}^{m} \left(\sum_{j=1}^{n} (c_{ij}x_{ij}) \right) \qquad (4.1)$$

---

* Een mogelijk alternatief voor het standaard simplex algorithme is het zgn. revised simplex algorithme ([4]). Voor computerberekening verdient deze methode soms de voorkeur. Beide methoden berusten echter op hetzelfde principe.

$$ST \quad \sum_{j=1}^{n} (x_{ij}) = S_i;\ i = 1,\ldots, m \tag{4.2}$$

$$\sum_{i=1}^{m} (x_{ij}) = D_j;\ j = 1,\ldots, n \tag{4.3}$$

$$x_{ij} \geq 0;\ i= 1,\ldots, m;\ j = 1,\ldots, n \tag{4.4}$$

met c_{ij}: de transportkosten per eenheid bij transport van aanbodpunt i naar vraagpunt j
D_j : de vraag van vraagpunt j
m : het aantal aanbodpunten i
n : het aantal vraagpunten j
S_i : het aanbod van aanbodpunt i
x_{ij}: het aantal eenheden dat van aanbodpunt i naar vraagpunt j getransporteerd wordt

Wil dit probleem een toelaatbare oplossing hebben dan zal moeten gelden:

$$\sum_{i=1}^{m} (S_i) = \sum_{j=1}^{n} (D_j) \tag{4.5}$$

Het totale aanbod zal even groot moeten zijn als de totale vraag, om aan alle beperkingen (4.2) en (4.3) te kunnen voldoen.

Een probleem waarbij het totale aanbod kleiner is dan de totale vraag kan volgens (4.1) t/m (4.4) geformuleerd worden door een fictief aanbodpunt i^* op te nemen met als aanbod het verschil tussen de totale vraag en het totale aanbod:

$$S_{i^*} = \sum_{j=1}^{n} (D_j) - \sum_{i=1}^{m} (S_i)$$

Door de transportkosten c_{i^*j} de waarde M te geven (een zeer groot getal) zal pas transport vanuit het aanbodpunt i^* plaatsvinden nadat alle overige aanbodpunten zijn uitgeput. Hierdoor kan de oplossing toch geïnterpreteerd worden.

Analoog kunnen ook problemen waarbij het totale aanbod groter is dan de totale vraag volgens als transportprobleem geformuleerd worden door een fictief vraagpunt j^* op te nemen met een vraag van:

$$D_{j^*} = \sum_{i=1}^{m} (S_i) - \sum_{j=1}^{n} (D_j)$$

Door de transportkosten c_{ij^*} de waarde 0 te geven heeft het opnemen van het fictieve vraagpunt j^* geen invloed op de optimale oplossing.

In de formulering volgens de vergelijkingen (4.1) t/m (4.4) heeft het transportprobleem twee specifieke kenmerken:

1. De coëfficiënten van de variabelen in de beperkingen hebben de waarde 0 of 1.

2. Het totale aanbod is gelijk aan de totale vraag.

Gebruik makend van deze twee kenmerken kan een oplossingsalgorithme geformuleerd worden dat veel efficiënter is dan het standaard simplex algorithme: het simplex algorithme voor transportproblemen. Dit algorithme wordt uitgebreid beschreven in [2] en [4] en wordt middels een voorbeeld toegelicht in bijlage IV.

Het simplex algorithme voor transportproblemen heeft een grotere efficiëntie dan het standaard simplex algorithme omdat de coëfficiëntenmatrix niet in iedere iteratiestap volledig behoeft te worden aangepast, zoals bij het standaard simplex algorithme. Slechts een deel van de posities in de matrix wordt volgens een bepaalde methodiek aangepast; dit beperkt het aantal rekenoperaties (zie bijlage IV). Daarnaast behoeven geen kunstmatige spelingsvariabelen te worden ingevoerd, waardoor de coëfficiëntenmatrix kleiner is dan bij het standaard simplex algorithme. In bijlage V is aangegeven hoe het transportprobleem, dat als voorbeeld is gebruikt in bijlage IV, geformuleerd moet worden om het met het standaard simplex algorithme op te kunnen lossen. Uit de bijlagen IV en V blijkt dat het tableau van het standaard simplex algorithme aanzienlijk meer posities telt dan het tableau van het simplex algorithme voor transportproblemen.

4.4 Het kostenmodel en het simplex algorithme voor transportproblemen

Het kostenmodel is in paragraaf 3.7 alsvolgt geformuleerd:

$$\min TCG = \sum_{i=1}^{n} \{A(i)d1(i)\} + \sum_{i=1}^{n} \{B(i)d2(i)\} + \sum_{i=1}^{n} \{ \sum_{s=1}^{S} \{E(i,s)d3(i,s)\}\} +$$

$$+ \sum_{i=1}^{n} \{ \sum_{j=1}^{m} \{F(i,j)qw(i,j)\}\} + \sum_{i=1}^{n} \{ \sum_{j=1}^{m} \{G(i,j)qb(i,j)\}\} +$$

$$+ \sum_{i=1}^{n} \{ \sum_{k=1}^{K} \{H(i,k)qx(i,k)\}\} \qquad (3.34)$$

$$\text{ST} \quad \sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} \geq a(j);\ j = 1,\ldots, m \qquad (3.35)$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - rq(i,0)d1(i) \geq 0 \qquad (3.36)$$

$$i = 1,\ldots, n$$

$$-\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} + QA.d1(i) \geq 0 \qquad (3.37)$$

$$i = 1,\ldots, n$$

$$-\sum_{j=1}^{m} \{qw(i,j)\} + QA.d2(i) \geq 0;\ i = 1,\ldots, n \qquad (3.38)$$

$$-\sum_{j=1}^{m} \{coef(i,s,j)qb(i,j)\} + QB(i,s)d3(i,s) \geq 0 \qquad (3.39)$$

$$i = 1,\ldots, n$$
$$s = 1,\ldots, S$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - qx(i,k) - kq(i,k)d4(i,k) \geq 0$$

$$i = 1,\ldots, n$$
$$k = 1,\ldots, K$$

$$(3.40)$$

$$-qx(i,k) + Q(i,k)d4(i,k) \geq 0;\ i = 1,\ldots, n \qquad (3.41)$$
$$k = 1,\ldots, K$$

$qw(i,j)$, $qb(i,j)$ en $qx(i,k) \geq 0$

$d1(i)$, $d2(i)$, $d3(i,s)$ en $d4(i,k)$: 0-1 variabelen

In paragraaf 4.2 is gesteld dat het kostenmodel het karakter heeft van een transportprobleem omdat er sprake is van aanbodpunten, van vraagpunten en van transport. In principe is dit juist; het kostenmodel is echter geen transportprobleem, zoals geformuleerd middels de vergelijkingen (4.1) t/m (4.4). Het verschilt op de volgende punten:

1. Hoewel er sprake is van aanbodpunten, is er geen sprake van een vooraf vastgesteld aanbod S_i per aanbodpunt. De oplossing van het model geeft de produktie van de winputten (die men als aanbod zou kunnen beschouwen).

2. Indien een winput in produktie genomen wordt is er een minimum produktie rq(0) vereist.

3. Er zijn twee transportwijzen; de oplossing van het model geeft de keuze voor de transportwijzen. Hierbij zal er bij transport vanuit een winput naar een bepaald vraagpunt transport per pijpleiding òf transport per vrachtwagen plaats kunnen vinden. Naar dit vraagpunt kan vanuit een andere winput wel transport plaatsvinden volgens de andere transportwijze. Evenzo kan vanuit één winput zowel transport per pijpleiding als transport per vrachtwagen plaatsvinden, maar dan naar verschillende vraagpunten.

4. Er is sprake van vaste kosten, zgn. "setup costs" die alleen optreden als de winput in produktie genomen wordt.

5. Er is sprake van setup costs die alleen optreden als een bepaalde transportwijze gekozen wordt.

6. Bij pijpleidingtransport is er sprake van setup costs die gekoppeld zijn aan de in gebruik zijnde pijpleidingtrajecten.

7. De aankoopkosten van de terreinen zijn niet rechtevenredig met de putproduktie. De kostenfunctie van de aankoopkosten is een spline functie, een geknikte lineaire functie.

Door deze verschillen met het transportprobleem is het niet mogelijk het kostenmodel in zijn huidige vorm als transportprobleem te formuleren en met het simplex algorithme voor transportproblemen op te lossen.

Door de formulering van het kostenmodel op een aantal punten aan te passen, overigens zonder hierbij de in paragraaf 3.2 genoemde uitgangspunten geweld aan te doen, is het waarschijnlijk wel mogelijk om het probleem met het simplex algorithme voor transportproblemen in combinatie met de branch-and-bound techniek op te lossen. Waarschijnlijk en niet zeker, omdat om zekerheid te krijgen het probleem daadwerkelijk geformuleerd en opgelost moet worden; dit viel echter buiten het kader van het afstudeeronderzoek.

Er wordt een aantal maatregelen voorgesteld om de formulering van het kostenmodel geschikt te maken voor oplossing met het simplex algorithme voor transportproblemen in combinatie met de branch-and-bound techniek. In de navolgende tekst corresponderen de nummers van de te nemen maatregelen met de nummers van de hiervoor aangegeven verschillen tussen de formulering van het kostenmodel en die van het transportprobleem.

1. Creëer voor iedere winput een vastgesteld aanbod S_i.

   Door het aanbod voor iedere winput gelijk te stellen aan

$$S_i = QA = \sum_{j=1}^{m} \{a(j)\}$$

   blijft het mogelijk dat vanuit één winput aan de vraag van alle vraagpunten voldaan wordt, een oplossing die in het oorspronkelijke kostenmodel ook mogelijk was. Er wordt dan niet voldaan aan de eis dat

$$\sum_{i=1}^{n} \{S_i\} = \sum_{j=1}^{m} \{D_j\},$$

   een eis die voor transportproblemen moet gelden. Dit is op te lossen door een fictief vraagpunt te creëren met als vraag

$$D_{m+1} = a(m+1) = \sum_{i=1}^{n} \{S_i\} - \sum_{j=1}^{m} \{a(j)\} = n*QA - QA = (n-1)*QA.$$

De kosten van transport naar dit fictieve vraagpunt moeten dan gelijk aan 0 gesteld worden.

2. Voorgesteld wordt om het probleem in eerste instantie op te lossen zonder de beperking, dat er een ondergrens voor de putproduktie rq(o) is indien de winput in produktie genomen wordt. Er kan daarna een vertakking plaatsvinden volgens de branch-and-bound techniek op de winputten die in produktie genomen worden en een produktie hebben die kleiner is dan rq(0). Hierbij wordt in de ene tak de put uit het probleem geëlimineerd en in de andere tak de eis dat q > rq(0) is gesteld. Dit wordt geïllusteerd middels fig. 15.

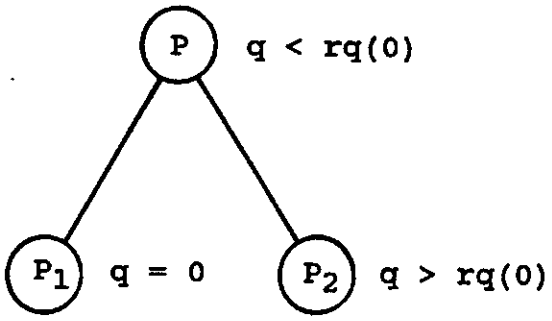


Fig. 15 Vertakking van de winput met q < rq(o) op de eis q > rq(0)

Het oplossen van het minimaliseringsprobleem met de beperking q > rq(0) komt bij maatregel 3 aan de orde.

3. De keuze voor de transportwijze kan in de formulering als transportprobleem worden opgenomen door het probleem te formuleren als een transshipment probleem. De theorie van de transshipment problemen wordt in de bestaande tekstboeken behandeld (o.a. [2]) en middels het volgende voorbeeld toegelicht:

Introduceer per winput P_i twee fictieve doorvoerpunten, een punt waaruit alleen transport per pijpleiding mogelijk is (L_i) en een punt waaruit alleen transport per vrachtwagen mogelijk is (A_i). De doorvoerpunten zijn hierbij zowel winput als vraagpunt. Dit wordt geïllustreerd middels fig. 16. Het vraagpunt V_f is het onder maatregel 1 besproken fictieve vraagpunt.

Het probleem wordt in fig. 17 in schemavorm weergegeven. De mogelijke transporten en het aanbod en de vraag zijn aangegeven. Voor de totale vraag v geldt:

$$v = v_1 + v_2$$

Zoals voorgesteld onder maatregel 1 is het aanbod van iedere winput gelijk aan de totale vraag v. Het fictieve vraagpunt V_f heeft dan ook een vraag v. De "x" geven aan dat transport van die winput of dat doorvoerpunt naar betreffend vraagpunt of doorvoerpunt niet mogelijk is. De "o" geven aan dat er wel transport mogelijk is. De "o* " zijn de transporten waarbij de transportkosten bij transport per pijpleiding of per vrachtwagen behoren. Het fictieve transport van bijvoorbeeld L_1 naar L_1 is in deze formulering te interpreteren als dat gedeelte van het werkelijke transport dat niet via L_1 gaat. Noem dit transport t. Dan gaat 2v - t wel via L_1 naar de vraagpunten V_1 en V_2.

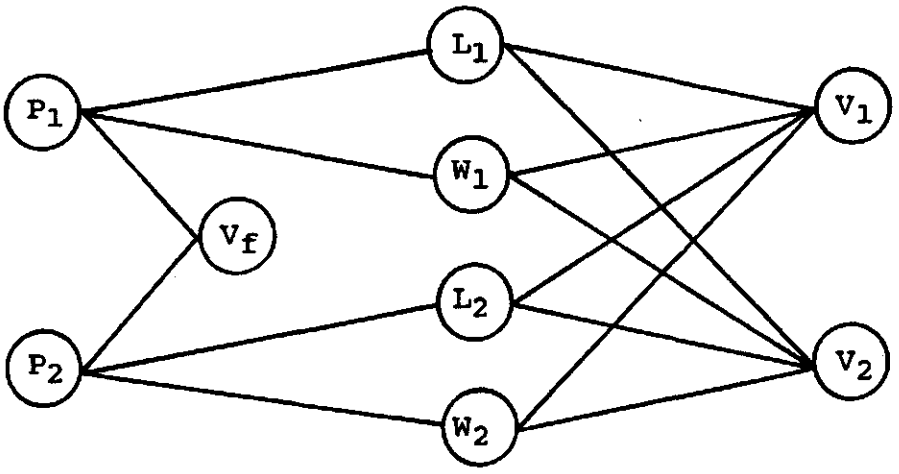


Fig. 16 Splitsen van een winput in twee doorvoerpunten met ieder één transportwijze (transshipment probleem)

		naar							
		L_1	W_1	L_2	W_2	V_1	V_2	V_f	aanbod
van	P_1	o	o	x	x	x	x	o	v
	P_2	x	x	o	o	x	x	o	v
	L_1	o	x	x	x	o*	o*	x	2v
	W_1	x	o	x	x	o*	o*	x	2v
	L_2	x	x	o	x	o*	o*	x	2v
	W_2	x	x	x	o	o*	o*	x	2v
	vraag	2v	2v	2v	2v	v_1	v_2	v	

Fig. 17 De keuze van de transportwijzen geformuleerd als transshipment probleem

Het op deze wijze geformuleerde probleem is een transportproblem en kan met het simplex algorithme voor transportproblemen worden opgelost.

Hoe de minimale produktie rq(0) (maatregel 2) nu in de formulering als transshipment probleem opgenomen kan worden opgenomen wordt middels het volgende voorbeeld geïllustreerd:

Realiseer een minimale produktie rq(0) van winput P_1 door het transport via L_1 minimaal m = rq(0) te laten zijn. Verlaag hiertoe de vraag van L_1 van 2v tot 2v - m en voer een fictief vraagpunt V_m in dat via alle doorvoerpunten kan worden voorzien, maar niet via L_1. Dit is in fig. 18 in schemavorm weergegeven.

		naar								
		L_1	W_1	L_2	W_2	V_1	V_2	V_f	V_m	aanbod
van	P_1	o	o	x	x	x	x	o	x	v
	P_2	x	x	o	o	x	x	o	x	v
	L_1	o	x	x	x	o*	o*	x	x	2v
	W_1	x	o	x	x	o*	o*	x	0	2v
	L_2	x	x	o	x	o*	o*	x	0	2v
	W_2	x	x	x	o	o*	o*	x	0	2v
vraag		2v-m	2v	2v	2v	v_1	v_2	v	m	

Fig. 18 Introductie van een fictief vraagpunt V_m om een minimale putproduktie m = rq(0) te realiseren voor winput P_1

Via L_1 worden de vraagpunten V_1 en V_2 nu minimaal voorzien van een hoeveelheid 2v - (2v - m) = m = rq(0).

Voor de andere doorvoerpunten kan dit op overeekomstige wijze gerealiseerd worden.

4. Door de setup costs is er sprake van een zgn. "fixed charge" probleem. Dit type problemen is eveneens met de branch-and-bound techniek op te lossen:

Los het probleem zonder setup costs op en vertak de winputten die geopend worden op de beslissing of de winput al dan niet geopend wordt. Introduceer in het eerste geval geen setup costs en in het tweede geval wel (fig. 19). De tak met setup costs is nu met het simplex algorithme voor transportproblemen op te lossen. In [18] wordt hiervoor een methode aangereikt; deze methode baseert zich op een aangepast selectiecriterium voor de toetredende basisvariabele in het simplex tableau voor

transportproblemen. In [18] wordt gesteld dat deze methode geschikt is voor een handmatige berekening van kleine problemen. Nader onderzoek moet uitwijzen of grotere problemen niet met een op deze methode gebaseerd computerprogramma kunnen worden opgelost.

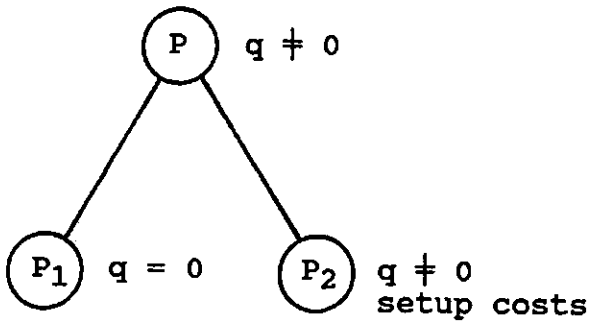


Fig. 19 Vertakking van de winput op basis van de setup costs

5. De setup costs die gekoppeld zijn aan de transportwijze kunnen in de formulering van het kostenmodel worden opgenomen door een combinatie van de maatregelen 3 en 4.

6. De pijpleidingtrajecten kunnen eveneens in het probleem worden opgenomen middels de formulering als transshipment probleem. De doorvoerpunten worden geïntroduceerd t.p.v. de splitsingen in het pijpleidingnetwerk. De setup costs voor de in gebruik zijnde pijpleidingtrajecten kunnen nu in de formulering van het kostenmodel worden opgenomen door de maatregelen 3. en 4. te combineren.

7. Het probleem van de geknikte kostenfunctie voor de aankoopkosten van de terreinen kan worden opgelost door een extra doorvoerpunt P_1' te introduceren dat middels een aantal transporttakken met de betreffende winput verbonden is (fig. 20). De takken stellen de verschillende produktieniveau's die gekoppeld zijn aan de verschillende kostenniveau's voor de aankoopkosten van de terreinen.

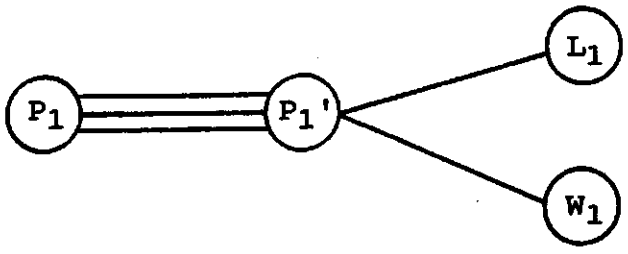


Fig. 20 Verschillende transporttakken die corresponderen met verschillende produktieniveau's

De conclusie is dat het ernaar uitziet dat het kostenmodel, geformuleerd als gemengd geheeltallig lineair programmeringsproblem, door een aantal maatregelen te nemen als transportproblem geformuleerd kan worden en dat het met het simplex algorithme voor transportproblemen in combinatie met de branch-and-bound techniek kan worden opgelost. Of deze methode ook geschikt is problemen van een omvang als die van het kostenmodel met het volledige proefgebied te kunnen oplossen, zal uit nader onderzoek moeten blijken. Opgemerkt kan worden dat de maatregelen inhouden dat een groot aantal fictieve doorvoerpunten wordt geïntroduceerd, hetgeen het aantal posities in het simplex tableau voor transportproblemen doet toenemen, waardoor de kans dat deze methode tot een oplossing leidt, afneemt.

5. MPSX

5.1 Inleiding

Zoals in hoofdstuk 4 is aangegeven, is het standaard softwarepakket MPSX (Mathematical Programming System eXtended, [19]) van IBM gebruikt het om het gemengd geheeltallige lineaire programmeringprobleem, waaruit het kostenmodel bestaat, op te lossen. Vanwege de geheeltalligheid van het probleem is hierbij gebruik gemaakt van de MIP-optie (Mixed Integer Programming, [20]). MPSX stond tot begin 1987 ter beschikking van de gebruikers van de Centrale Digitale Installatie van het Rekencentrum van de TUD.

Het lag in de bedoeling om met MPSX het kostenmodel op te lossen voor het volledige proefgebied, zoals beschreven in bijlage II. Het minimaliseringsprobleem dat hierbij onstaat, is echter dusdanig omvangrijk dat oplossing met MPSX niet mogelijk is. Dit kan worden aangetoond middels de volgende berekening:

Het probleem omvat de volgende aantallen beperkingen en variabelen:

beperkingen (3.35 t/m 3.41): aantal:

$$\sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} \geq a(j);\ j = 1,\ldots, m \qquad m$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - rq(i,0)d1(i) \geq 0;\ i = 1,\ldots, n \qquad n$$

$$-\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} + QA.d1(i) \geq 0;\ i = 1,\ldots, n \qquad n$$

$$-\sum_{j=1}^{m} \{qw(i,j)\} + QA.d2(i) \geq 0;\ i = 1,\ldots, n \qquad n$$

$$-\sum_{j=1}^{m} \{coef(i,s,j)qb(i,j)\} + QB(i,s)d3(i,s) \geq 0; \qquad n.S$$

$$i = 1,\ldots, n \quad s = 1,\ldots, S$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - qx(i,k) - kq(i,k)d4(i,k) \geq 0; \qquad n.K$$

$$i = 1,\ldots, n \quad k = 1,\ldots, K$$

$$-qx(i,k) + Q(i,k)d4(i,k) \geq 0;\ i = 1,\ldots, n \quad k = 1,\ldots, K \qquad n.K$$

Totaal (3+S+2K).n+m beperkingen.

variabelen:	aantal:
qw(i,j)	n.m
qb(i,j)	n.m
qx(i,k)	n.K
d1(i)	n
d2(i)	n
d3(i,s)	n.S
d4(i,k)	n.K

Totaal (2+2m+2K+S).n variabelen.

Zoals uit bijlage II blijkt, bestaat het proefgebied uit 19 potentiële winputten en 47 vraagpunten en telt het leidingennet 86 trajecten (n=19, m=47 en S=86). Wanneer het aantal spline functie intervallen op 5 gesteld wordt (K=5) is er sprake van 1928 beperkingen en 3648 variabelen (exclusief de spelingsvariabelen). De probleemmatrix van MPSX neemt dan 7033344 posities in beslag (1928x3648) hetgeen bij een geheugengrootte van 4 bytes per positie (real variabelen, floating point) een intern geheugenbeslag geeft van 27474 kBytes. Dit ging de grootte van het interne geheugen van de Rekencentrum-computer te boven.

Voor het volledige probleem was het dus niet mogelijk om met MPSX tot een oplossing te komen. Wel is een deelprobleem met MPSX opgelost. Hiermee is een oplossing verkregen waaraan de oplossing van het ICW-oplossingsalgorithme voor dit deelprobleem getoetst kon worden (zie paragraaf 7.12).

5.2 Gebruikte programma's

Bij het oplossen van het deelprobleem met MPSX is gebruik gemaakt van de volgende programma's:

1. **INPUT;**: met dit in het kader van het onderzoek in FORTRAN 77 ([6]) geschreven programma worden de benodigde gegevens ingelezen, worden de coëfficienten van het kostenmodel berekend en worden de ingelezen en berekende gegevens uitgevoerd op een voor de invoergenerator GEN80 voorgeschreven wijze. Een listing van het programma INPUT en een verklaring van de betekenis van de in het programma gebruikte variabelen zijn opgenomen in bijlage VI.

2. **GEN80**; het programma GEN80 ([9]) is een invoergenerator voor MPSX, die door de Onderafdeling der Wiskunde en Informatica van de Technische Hogeschool Delft is ontwikkeld. De invoer voor GEN80 bestaat uit het op een voor GEN80 voorgeschreven wijze weergeven kostenmodel en uit de uitvoer van het programma INPUT. De uitvoer van GEN80 bestaat uit de invoermatrix voor MPSX. In bijlage VII is de invoer van GEN80 voor het deelprobleem weergegeven. De uitvoer van GEN80 bevat geen zinvolle informatie en is daarom niet weergegeven.

3. MPSX; het reeds besproken softwarepakket van IBM voor gemengd geheeltallige lineaire programmering. In bijlage VIII is aangegeven met welke opties MPSX gedraaid is.

5.3 Het deelprobleem

Het programma MPSX is gedraaid voor een deelprobleem, bestaande uit de potentiële winputten 4 en 18 en de vraagpunten 8, 30, 39, 40 en 41. In fig. 21 is het deelgebied van dit deelprobleem gemarkeerd als onderdeel van het volledige proefgebied. Het deelprobleem is hierbij zodanig gekozen, dat er zowel transport per vrachtwagen als per pijpleiding plaatsvindt. Dit maakt de oplossing geschikt om er de uitkomsten van het ICW-algorithme mee te toetsen.

In bijlage VI zijn de invoergegevens van het deelprobleem opgenomen. De uitkomsten van MPSX voor het deelprobleem zijn opgenomen in bijlage VIII.

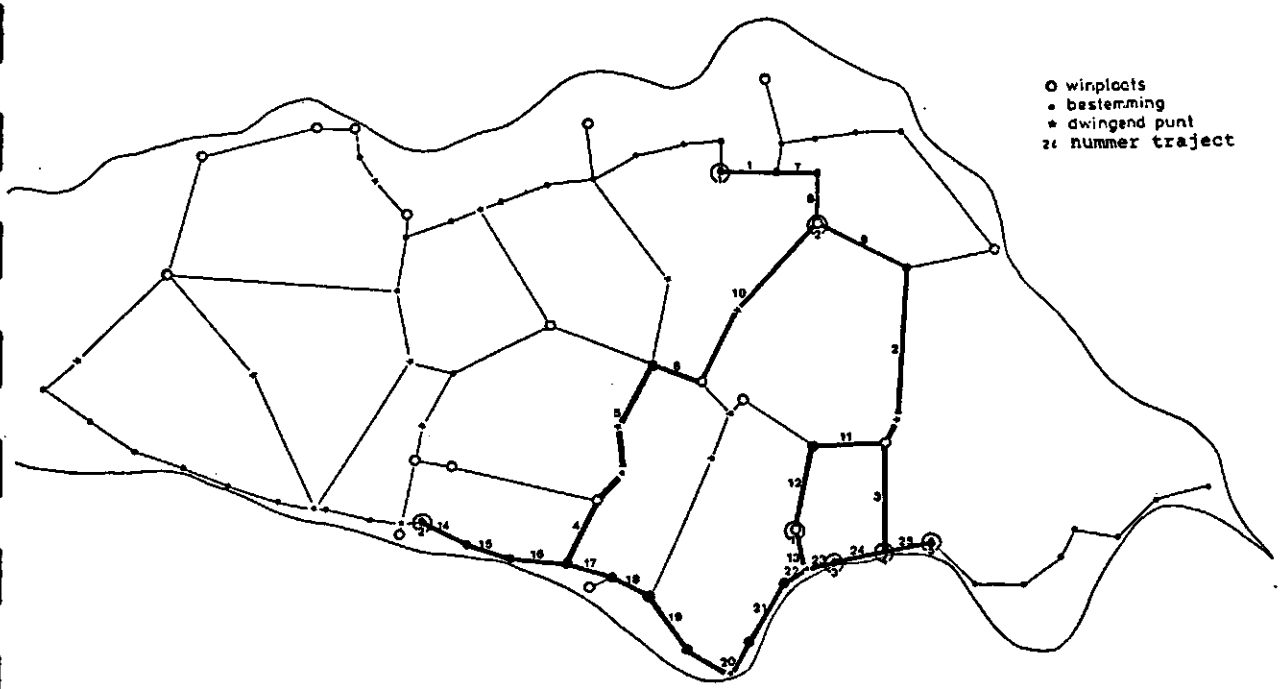


Fig. 21 Deelgebied als onderdeel van het volledige proefgebied (nummering volgens bijlage VI)

5.4 Conclusies

De voornaamste conclusie is dat het kostenmodel met MPSX kan worden opgelost, maar dat MPSX niet geschikt is om gemengd geheeltallige lineariseringsproblemen op te lossen van een omvang als die van het kostenmodel voor het volledige proefgebied.

Daarnaast kan geconcludeerd worden dat het kostenmodel voor het deelprobleem goed functioneert. De resultaten van MPSX voor het deelprobleem zijn door middel van een handberekening gecontroleerd (bijlage VIII) en de uitkomsten hiervan komen overeen met de resultaten van MPSX.

6. ICW-oplossingsalgorithme en computerprogramma

6.1 Inleiding

Het kostenmodel is het volgens de vergelijkingen (3.34) t/m (3.41) geformuleerde minimaliseringsprobleem. Dit minimaliseringsprobleem moet worden opgelost voor het proefgebied. Het proefgebied (zie bijlage II) bestaat uit 19 potentiële zandwinputten en 47 vraagpunten, welke onderling verbonden worden gedacht door een pijpleidingennetwerk met 86 trajecten. Wanneer er 5 spline functie intervallen gedefinieerd worden omvat het minimaliseringsprobleem 1928 beperkingen en 3648 variabelen (exclusief de spelingsvariabelen), waarvan 1881 continue en 1767 discrete- of dummievariabelen. Door de omvang van het probleem zullen de gebruikelijke algorithmen voor gemengd geheeltallige lineaire programmering een zeer groot aantal iteraties vergen om het optimum te bereiken. Bovendien zal het geheugenbeslag van de matrix, nodig om het probleem met de gebruikelijke algorithmen met behulp van de computer op te lossen, veelal te groot zijn. Dit wordt bevestigd door het gebruik van het standaard softwarepakket MPSX (zie het voorgaande hoofdstuk).

Vandaar dat door het ICW een eigen algorithme voor gemengd geheeltallige lineaire programmering ontwikkeld is. Dit algorithme wordt beschreven in [12]. Voor de theoretische achtergronden van het algorithme verwezen wordt naar bestaande tekstboeken.

Een onderdeel van het algorithme is de oplossing van het duale probleem van het primale probleem (3.34) t/m (3.41). Omdat dit duale probleem een groot aantal beperkingen omvat (zie paragraaf 6.2) heeft het ICW het algorithme op dit punt naderhand aangepast. Deze aanpassing is niet opgenomen in [12] en wordt daarom in dit hoofdstuk beschreven.

In dit hoofdstuk wordt het oplossingsalgorithme in relatie tot het minimaliseringsprobleem beschreven. Daarnaast wordt het computerprogramma beschreven, dat op basis van het oplossingsalgorithme ontwikkeld is. Gedurende het onderzoek heeft het programma een zekere ontwikkeling doorgemaakt. In dit hoofdstuk wordt de laatste versie van het programma beschreven. De ontwikkeling en het testen van het programma komen aan de orde in hoofdstuk 7.

6.2 Toepassen van het oplossingsalgorithme op het minimaliseringsprobleem

Het in [12] en [13] geformuleerde algorithme voor gemengd geheeltallige lineaire programmering wordt gebruikt om het probleem (3.34) t/m (3.41) op te lossen. Onder toevoeging van de spelingsvariabelen kan het probleem volgens (6.1) geformuleerd worden:

$$\min \sum_{i=1}^{n} \{A(i)d1(i)\} + \sum_{i=1}^{n} \{B(i)d2(i)\} + \sum_{i=1}^{n} \sum_{s=1}^{S} \{E(i,s)d3(i,s)\} +$$

$$+ \sum_{i=1}^{n} \sum_{j=1}^{m} \{F(i,j)qw(i,j)\} + \sum_{i=1}^{n} \sum_{j=1}^{m} \{G(i,j)qb(i,j)\} +$$

$$+ \sum_{i=1}^{n} \sum_{k=1}^{K} \{H(i,k)qx(i,k)\} \qquad (6.1)$$

$$\text{ST} \quad \sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} - YS(0,j) = a(j);\ j = 1,\ldots, m$$

$$\sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} - YS(1,i,1) = rq(i,0)d1(i);\ j = 1,\ldots, m$$

$$\sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} - YS(1,i,2) = QA.d1(i);\ j = 1,\ldots, m$$

$$\sum_{j=1}^{m} \{qw(i,j)\} + YS(2,i) = QA.d2(i);\ i = 1,\ldots, n$$

$$\sum_{j=1}^{m} \{coef(i,s,j)qb(i,j)\} + YS(3,i,s) = QB(i,s)d3(i,s);$$

$$i = 1,\ldots, n$$
$$s = 1,\ldots, S$$

$$\sum_{j=1}^{m} \{qb(i,j) + qw(i,j)\} - qx(i,k) - YS(4,i,k) = kq(i,k)d4(i,k);$$

$$i = 1,\ldots, n$$
$$k = 1,\ldots, K$$

$$qx(i,k) + YS(5,i,k) = Q(i,k)d4(i,k);\ i = 1,\ldots, n$$
$$k = 1,\ldots, K$$

$qw(i,j)$, $qb(i,j)$, $qx(i,k)$ en $YS \geq 0$

$d1(i)$, $d2(i)$, $d3(i,s)$ en $d4(i,k)$: 0-1 variabelen

De betekenis van de in (6.1) gebruikte variabelen m.u.v. de spelingsvariabelen wordt verklaard in hoofdstuk 3. De spelingsvariabelen zijn: $YS(0,j)$, $YS(1,i,1)$, $YS(1,i,2)$, $YS(2,i)$, $YS(3,i,s)$, $YS(4,i,k)$ en $YS(5,i,k)$.

In het oplossingsalgorithme wordt gebruik gemaakt van het duale probleem van (6.1). Dit duale probleem kan alsvolgt geformuleerd worden:

$$\min_{d} \max_{u} \sum_{j=1}^{m} \{u(0,j)a(j)\} + \sum_{i=1}^{n} [\{A(i) + rq(i,0)u(1,i,1) -$$

$$- QA.u(1,i,2))\}d1(i) + \{B(i) - QA.u(2,i)\}d2(i)] +$$

$$+ \sum_{i=1}^{n} \sum_{s=1}^{S} [\{(E(i,s) - QB(i,s)u(3,i,s)\}d3(i,s)] +$$

$$+ \sum_{i=1}^{n} \sum_{k=1}^{K} [\{(kq(i,k)u(4,i,k) - Q(i,k)u(5,i,k)\}d4(i,k)] \qquad (6.2)$$

$$\text{ST } u(0,j) + u(1,i,1) - u(1,i,2) - u(2,i) + \sum_{k=1}^{K} \{u(4,i,k)\} +$$

$$+ XS(1,i,j) = F(i,j); \; i = 1,\ldots, n \quad j = 1,\ldots, m$$

$$u(0,j) + u(1,i,1) - u(1,i,2) - \sum_{s=1}^{S} \{coef(i,s,j)u(3,i,s)\} +$$

$$+ \sum_{k=1}^{K} \{u(4,i,k)\} + XS(2,i,j) = G(i,j); \; i = 1,\ldots, n \quad j = 1,\ldots, m$$

$$-u(4,i,k) - u(5,i,k) + XS(3,i,k) = H(i,k); \; i = 1,\ldots, n \quad k = 1,\ldots, K$$

$$u, XS \geq 0$$

$$d1(i), d2(i), d3(i,s) \text{ en } d4(i,k): 0\text{-}1 \text{ variabelen}$$

De duale variabelen corresponderen hierbij met de primale variabelen zoals aangegeven wordt in tabel 5:

primaal	duaal	primaal	duaal
qw(i,j)	XS(1,i,j)	YS(1,i,2)	u(1,i,2)
qb(i,j)	XS(2,i,j)	YS(2,i)	u(2,i)
qx(i,k)	XS(3,i,k)	YS(3,i,s)	u(3,i,s)
YS(0,j)	u(0,j)	YS(4,i,k)	u(4,i,k)
YS(1,i,1)	u(1,i,1)	YS(5,i,k)	u(5,i,k)

Tabel 5 Relatie tussen de primale en de duale variabelen

De opzet van het oplossingsalgorithme is de volgende:

1. Bepaal voor het duale probleem (6.2) een toelaatbare oplossing voor de dummievariabelen d. Dit is de startoplossing voor de dummievariabelen.

2. Bepaal de optimale oplossing van het duale probleem voor de duale variabelen u en XS voor gegeven waarden van d.

3. Bepaal de optimale oplossing van het duale probleem voor d voor gegeven waarden van u en XS.

Start het algorithme met stap 1. Herhaal vervolgens de stappen 2 en 3 totdat de optimale oplossing bereikt is. De optimale oplossing is bereikt indien de in de stappen 2. en 3. gevonden optima aan elkaar gelijk zijn. Transformeer tenslotte de optimale oplossing van het duale probleem naar de optimale oplossing van het primale (oorspronkelijke) probleem.

Het oplossingsalgorithme is schematisch weergegeven in fig. 22.

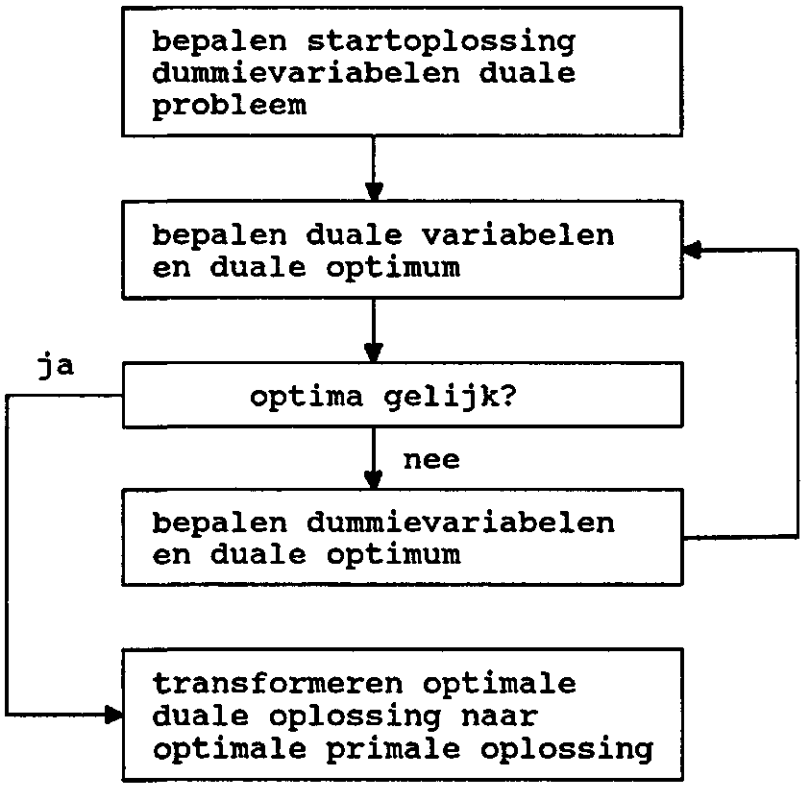


Fig. 22 Schematische weergave van het oplossingsalgorithme

De gegeven waarden van de dummievariabelen zijn bij de eerste iteratie van het oplossingsalgorithme afkomstig uit stap 1 en bij volgende iteraties uit stap 3. De optimale oplossing is bereikt wanneer de waarden van de optima, zoals gevonden in de stappen 2 en 3, aan elkaar gelijk zijn.

Het in stap 2 op te lossen duale probleem is een maximaliserings-probleem in u en XS. Wanneer er 5 spline functie intervallen gedefinieerd worden omvat dit probleem voor het proefgebied 1881 beperkingen. Daarom wordt in stap 2 niet het duale probleem

opgelost maar het primale probleem, zijnde het duale van het duale probleem. Na vereenvoudiging omvat dit primale probleem maximaal 66 beperkingen. Deze benadering heeft de volgende aanpassingen van het oplossingsalgorithme tot gevolg:

Aanpassing stap 1:

Het wordt zinvol geacht een toelaatbare startoplossing voor de dummievariabelen te bepalen die zo dicht mogelijk bij de optimale oplossing ligt. In stap 2 wordt niet het duale, maar het primale probleem voor gegeven waarden van de dummievariabelen opgelost. Vandaar dat in stap 1 de dichtstbijzijnde oplossing voor de dummievariabelen wordt bepaald voor het primale probleem.

Aanpassing stap 2:

In stap 2 wordt het vereenvoudigde primale probleem voor gegeven waarden van de dummievariabelen opgelost. De oplossing van het vereenvoudigde primale probleem wordt omgezet in de oplossing van het volledige primale probleem en vervolgens getransformeerd naar de oplossing van het duale probleem.

Na invoering van deze aanpassingen bestaat het oplossingsalgorithme uit de volgende stappen:

1. Bepaal langs analytische weg voor het primale probleem (6.1) de dichtstbijzijnde startoplossing voor de dummievariabelen.

2. Vereenvoudig het primale probleem en bepaal voor gegeven waarden van de dummievariabelen de optimale oplossing voor de continue variabelen met het revised simplex algorithme. De gegeven waarden van de dummievariabelen zijn bij de eerste iteratie van het oplossingsalgorithme afkomstig uit stap 1 en bij volgende iteraties uit stap 5. Zet vervolgens de optimale oplossing van het vereenvoudigde primale probleem om in de optimale oplossing van het volledige primale probleem.

3. Vergelijk de optima, zoals gevonden in de stappen 2 en 5. Zijn deze optima aan elkaar gelijk, dan is de optimale oplossing gevonden. Zo niet, dan wordt de procedure vervolgd met stap 4 (in de eerste iteratie van het algorithme is er nog geen optimale oplossing gevonden in stap 5; stap 3 heeft dan geen betekenis).

4. Transformeer de optimale oplossing van het primale probleem voor de continue variabelen naar die van het duale probleem. Deze getransformeerde oplossing heeft de vorm van een constante en coëfficiënten voor de dummievariabelen. Deze getransformeerde oplossing vormt een vergelijking die gebruikt wordt in stap 5.

5. Voeg de in stap 4 verkregen vergelijking toe aan de vergelijkingen die in voorgaande iteraties in stap 4 verkregen zijn en bepaal met behulp van de zoekmethode voor 0-1 variabelen de optimale oplossing voor de dummievariabelen van het stelsel vergelijkingen. Herhaal vervolgens de procedure vanaf stap 2.

In fig. 23 is het aangepaste oplossingsalgorithme schematisch weergegeven.

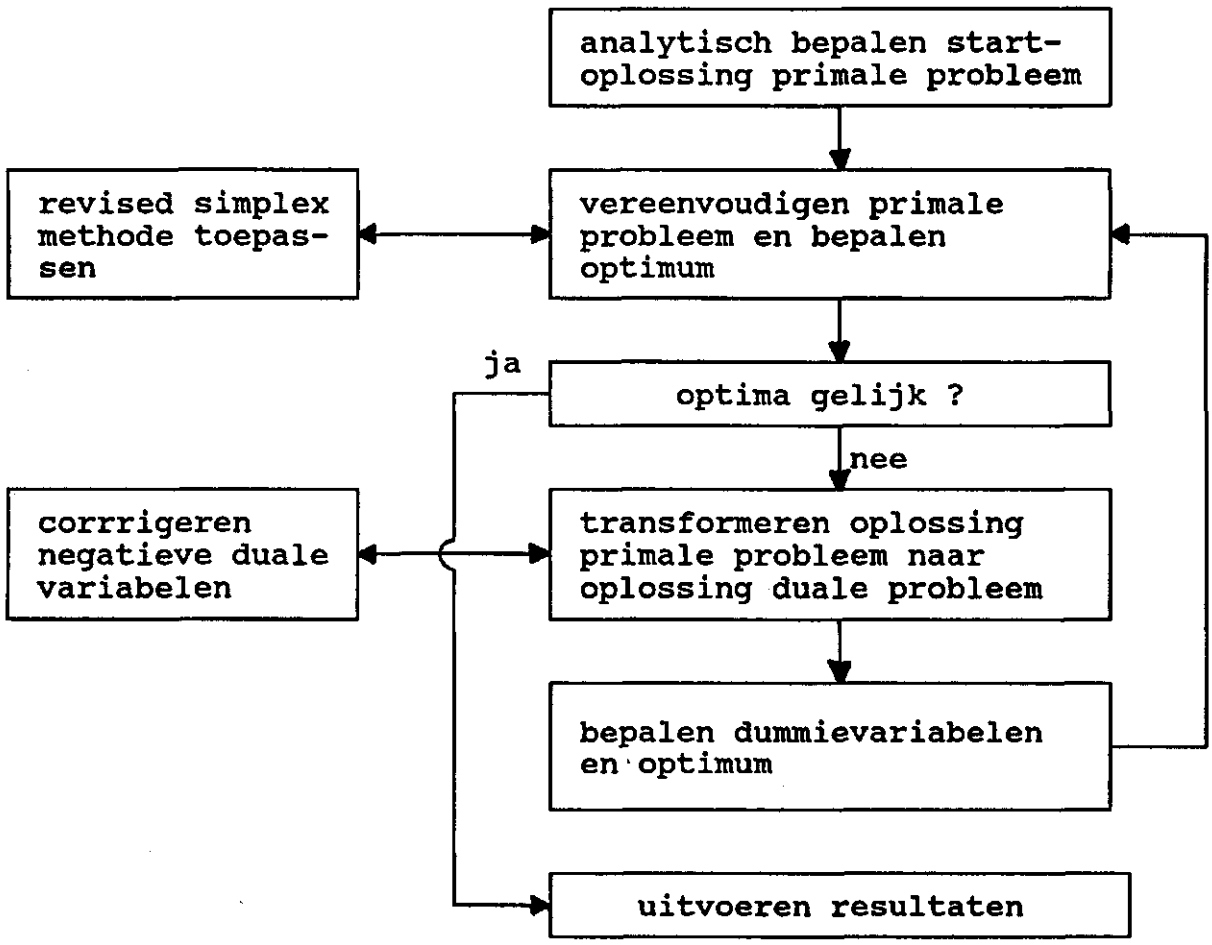


Fig. 23 Schematische weergave van het aangepaste oplossingsalgorithme

6.3 Van oplossingsalgorithme naar computerprogramma

Op basis van het oplossingsalgorithme, zoals beschreven in de voorgaande paragraaf, is een computerprogramma ontwikkeld. In het vervolg van dit hoofdstuk wordt de laatste versie van het programma besproken. De ontwikkeling en het testen van het programma komen in hoofdstuk 7 aan de orde.

Het computerprogramma, dat geschreven is in FORTRAN 77 ([6]), bestaat uit de volgende onderdelen: het hulpprogramma MODIFY, het hoofdprogramma MAIN en de subroutines RECALCO, ANSOL, PRISOL, RESIM, TRANS, CORRECT en ZEON. De functie van deze programma-onderdelen en hun relatie tot het oplossingsalgorithme is de volgende:

MODIFY : In dit hulpprogramma worden de pijpleidingnetwerken vereenvoudigd en wordt de combinatiematrix comb(j,jh) samengesteld. Het wordt voorafgaande aan het programma MAIN gedraaid en maakt geen deel uit van het iteratieproces van het oplossingsalgorithme.

MAIN : Het hoofdprogramma MAIN bestuurt het iteratieproces. In het programma worden verschillende subroutines aangeroepen en wordt stap 3 van het algorithme uitgevoerd.

RECALCO: In de subroutine RECALCO worden de coëfficiënten van het kostenmodel berekend. De subroutine wordt eenmalig aangeroepen vanuit MAIN.

ANSOL : In deze subroutine wordt langs analytische weg voor het primale probleem de dichtstbijzijnde oplossing voor de dummievariabelen bepaald. ANSOL komt overeen met stap 1 van het oplossingsalgorithme en wordt eenmalig aangeroepen vanuit MAIN.

PRISOL : De subroutine PRISOL wordt aangeroepen vanuit MAIN. In de subroutine wordt het primale probleem vereenvoudigd, wordt een startbasis en een inverse matrix voor de revised simplex methode bepaald en wordt de subroutine RESIM aangeroepen om het vereenvoudigde primale probleem op te lossen. De optimale oplossing van het vereenvoudigde probleem wordt omgezet in die van het volledige primale probleem. PRISOL en RESIM komen samen overeen met stap 2 van het oplossingsalgorithme.

RESIM : In deze subroutine, die wordt aangeroepen vanuit de subroutine PRISOL, wordt het vereenvoudigde primale probleem opgelost m.b.v. de revised simplex methode.

TRANS : De subroutine TRANS wordt aangeroepen vanuit MAIN. In de subroutine wordt de oplossing van het primale probleem, zoals bepaald in de subroutines PRISOL en RESIM, getransformeerd naar de oplossing van het duale probleem. Hierbij kan het noodzakelijk zijn om de subroutine CORRECT aan te roepen. TRANS en CORRECT komen samen overeen met stap 4 van het oplossingsalgorithme.

CORRECT: De subroutine CORRECT wordt zonodig aangeroepen vanuit TRANS om gevonden waarden voor de duale variabelen te corrigeren.

ZEON : In deze subroutine, die aangeroepen wordt vanuit MAIN, wordt door middel van een zoekmethode voor 0-1 variabelen de optimale oplossing voor de dummievariabelen van het duale probleem bepaald. De subroutine ZEON komt overeen met stap 5 van het oplossingsalgorithme.

In de paragrafen 6.4 t/m 6.10 worden genoemde programmaonderdelen nader besproken. In de bijlagen IX t/m XVII zijn de listings van

deze programmaonderdelen opgenomen en wordt de betekenis van de gebruikte variabelen verklaard.

In fig. 24 wordt het computerprogramma schematisch weergegeven.

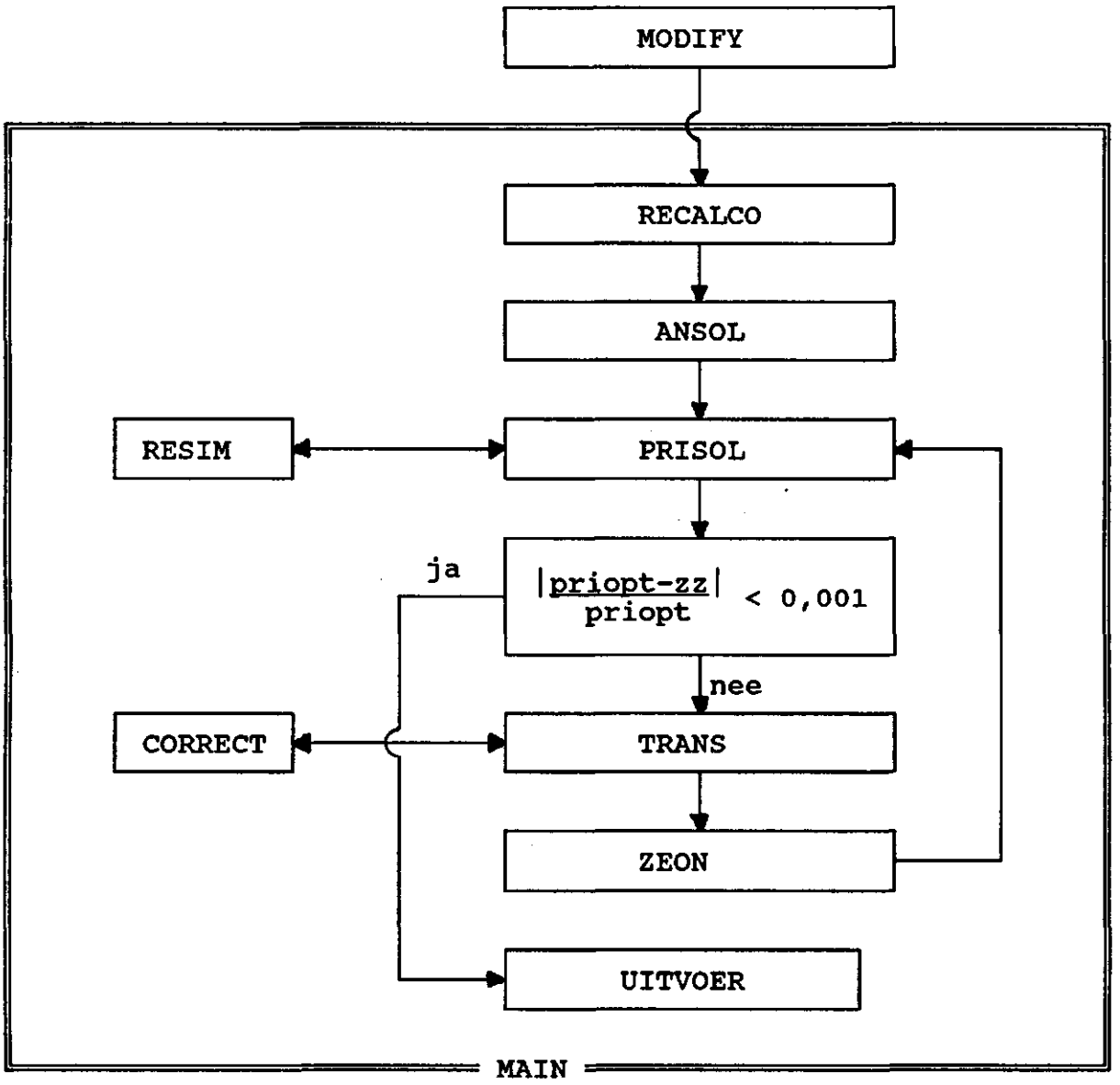


Fig. 24 Schematische weergave van het computerprogramma

6.4 Vereenvoudigen van de pijpleidingnetten en samenstellen van de combinatiematrix, het hulpprogramma MODIFY

De twee taken van het hulpprogramma MODIFY (**MODIFY** pipeline network) zijn het vereenvoudigen van de pijpleidingnetten en het samenstellen van de combinatiematrix comb(j,jh).

Vereenvoudigen van de pijpleidingnetten

1. Bepalen afzonderlijke leidingnetten per winput

In de oorspronkelijke opzet zijn de leidingnetten van de verschillende winputten weergegeven als één groot netwerk dat alle winputten van het proefgebied met alle vraagpunten verbindt. De 0-1 matrix coef(i,s,j) geeft hierbij aan welke trajecten* deel uitmaken van de leidingnetten van de verschillende winputten. De coëfficiënt coef(i,s,j) heeft de waarde 1 wanneer traject s deel uitmaakt van het leidingtracé dat winput i met vraagpunt j verbindt en de waarde 0 wanneer dit niet het geval is. Wanneer een traject in de leidingnetten van verschillende winputten voorkomt, wil dit niet zeggen dat het transport van zand vanuit die winputten door hetzelfde pijpleidingdeel zal voeren. In dat geval zal zo'n traject bestaan uit een aantal parallelle pijpleidingdelen.

De oorspronkelijke opzet is niet efficiënt omdat er in het kostenmodel per winput een groot aantal variabelen zonder reële betekenis wordt geïntroduceerd (nl. de 0-1 variabelen, die corresponderen met trajecten die geen deel uitmaken van de leidingtracé's die betreffende winput met alle vraagpunten verbinden, en dus voor alle j de waarde 0 hebben). Vandaar dat voor een opzet is gekozen, waarbij per winput een afzonderlijk leidingennet wordt bepaald, waarin alleen die trajecten voorkomen die daadwerkelijk deel uitmaken van de leidingtracé's die de winput met de verschillende vraagpunten verbinden. De consequentie van deze opzet is dat het aantal trajecten per winput nu verschillend is en dat daardoor de nummering van de parallelle leidingdelen onderling verschilt.

2. Samenvoegen van trajecten

In het leidingennet van een winput kunnen trajecten voorkomen die naar exact dezelfde vraagpunt(en) leiden. Deze trajecten kunnen zonder bezwaar samengevoegd worden. De lengte van het dan ontstane traject wordt hierbij gelijk aan de som van de lengten van de oorspronkelijke trajecten en de nummering van de trajecten wordt aangepast. Met dit samenvoegen van trajecten vermindert het aantal trajecten per winput en dus het aantal variabelen in het kostenmodel.

Samenstellen van de combinatiematrix comb(j,jh)

In de oorspronkelijke opzet van de subroutine ZEON werden de te bepalen dummievariabelen van het duale probleem als onderling onafhankelijk beschouwd. Aangenomen werd dat de onderlinge afhankelijkheid, die er tussen bepaalde dummievariabelen bestaat, tot uitdrukking zou komen in de in de subroutine TRANS bepaalde coëfficiënten van de dummievariabelen gam(t,j). Dit blijkt echter

---

* Onder een traject wordt verstaan een deel van het leidingennet dat aan begin en einde begrensd wordt door een winput, een splitsing of een vraagpunt.

niet te gelden (zie paragraaf 7.10), met ontoelaatbare oplossingen voor de dummievariabelen als gevolg*. Daarom is een nieuwe versie van ZEON ontwikkeld, waarin de onderlinge afhankelijkheid van de dummievariabelen is verwerkt. In ZEON wordt gebruik gemaakt van de combinatiematrix comb(j,jh), waarin de onderlinge afhankelijkheid van de dummievariabelen is vastgelegd.

Tussen de dummievariabelen bestaat de volgende onderlinge afhankelijkheid:

dummievariabele:	afhankelijke dummievariabelen:
d1(i)	geen
d2(i)	d1(i)
d3(i,s)	d1(i) en die d3(i,s), waarvan de trajecten s naar één of meer dezelfde vraagpunten leiden als traject s
d4(i,k)	d1(i) en die d4(i,k), waarvoor geldt k < k

In bijlage IX is een listing van het hulpprogramma MODIFY opgenomen en wordt de betekenis van de in het programma gebruikte variabelen verklaard.

6.5 Besturen van het computerprogramma, het hoofdprogramma MAIN

Het hoofdprogramma MAIN (MAIN program) bestuurt het computerprogramma. Dit houdt in het uitvoeren van de volgende taken:

- Een teller bijhouden die het aantal iteratiestappen registreert. Wanneer het aantal doorlopen iteratiestappen een vooraf vastgesteld maximum overschrijdt, wordt de uitvoering van MAIN gestopt.
- Het aanroepen van de verschillende subroutines.
- Het uitvoeren van de volgende toets:

$$\frac{|\text{priopt} - \text{zz}|}{\text{priopt}} < 0.001$$

  . priopt: de optimale oplossing van het primale probleem voor gegeven waarden van de dummievariabelen, berekend in de subroutine PRISOL [f]
  zz : de optimale oplossing van het primale probleem voor gegeven waarden van de continue variabelen, berekend in de subroutine ZEON [f]
  . Indien priopt en zz aan elkaar gelijk zijn, dan is de optimale oplossing van het minimaliseringsprobleem gevonden.
  . Reden om niet de toets priopt = zz uit te voeren is dat de mogelijkheid zich voor kan doen dat door afrondingsfouten bij de computerberekening niet aan deze toets voldaan wordt, terwijl de optimale oplossing wel gevonden is. De gekozen toets staat een verschil toe tussen priopt en zz van 0,1 %.

---

* Een voorbeeld van een ontoelaatbare oplossing is: d1(i)=0 en d2(i)=1, d.w.z. er is transport van zand per vrachtwagen vanuit winput i, terwijl deze winput niet in produktie genomen is.

- Het uitvoeren van de gebruikte parameters, het nummer van de iteratiestap en de in die iteratiestap gevonden minima en van de optimale oplossing.

De opbouw van het volledige programma, zoals dit door het hoofdprogramma MAIN bestuurd wordt, is in schemavorm weergegeven in fig. 24. In bijlage X is een listing van MAIN opgenomen en wordt de betekenis van de in het programma gebruikte variabelen verklaard.

6.6 Berekenen van de coëfficiënten van het kostenmodel, de subroutine RECALCO

In de subroutine RECALCO (REad and CALculate COefficients) worden de gegevens van het proefgebied ingelezen en de coëfficiënten van het kostenmodel berekend. Bij het berekenen van de spline functie coëfficiënten is uitgegaan van de relatieve afwijking ρ, hetgeen een arbitraire keuze is. In bijlage XI is een listing van RECALCO opgenomen en wordt de betekenis van de in de subroutine gebruikte variabelen verklaard.

6.7 Bepalen van een startoplossing voor de dummievariabelen, de subroutine ANSOL

Het wordt zinvol geacht om als startoplossing voor de dummievariabelen van het primale probleem een oplossing te kiezen die zo dicht mogelijk bij de optimale oplossing ligt. Deze startoplossing wordt analytisch bepaald in de subroutine ANSOL (ANalytical SOLution). In deze paragraaf wordt de in ANSOL gehanteerde methode beschreven; een listing van de subroutine ANSOL en een verklaring van de betekenis van de in ANSOL gebruikte variabelen is opgenomen in bijlage XII.

Wanneer voor het primale probleem (3.34) t/m (3.41) de dummievariabelen d1 - d4 continue worden verondersteld in plaats van geheeltallig en dus de beperkingen "d1 - d4 : 0-1 variabelen" vervangen worden door de beperkingen "$0 \leq$ d1 - d4 ≤ 1", dan kan het primale probleem alsvolgt geschreven worden:

$$\min \sum_{i=1}^{n} \{A(i)d1(i)\} + \sum_{i=1}^{n} \{B(i)d2(i)\} + \sum_{i=1}^{n} \sum_{s=1}^{S} \{E(i,s)d3(i,s)\} +$$

$$+ \sum_{i=1}^{n} \sum_{j=1}^{m} \{F(i,j)qw(i,j)\} + \sum_{i=1}^{n} \sum_{j=1}^{m} \{G(i,j)qb(i,j)\} +$$

$$+ \sum_{i=1}^{n} \sum_{k=1}^{K} \{H(i,k)qx(i,k)\} \qquad (6.3)$$

$$\text{ST} \quad \sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} - a(j) \geq 0; \; j = 1,\ldots, m$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - rq(i,0)d1(i) \geq 0; \; i = 1,\ldots, n$$

$$-\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} + QA.d1(i) \geq 0; \; i = 1,\ldots, n$$

$$-\sum_{j=1}^{m} \{qw(i,j)\} + QA.d2(i) \geq 0; \; i = 1,\ldots, n$$

$$-\sum_{j=1}^{m} \{coef(i,s,j)qb(i,j)\} + QB(i,s)d3(i,s) \geq 0; \; i = 1,\ldots, n \quad s = 1,\ldots, S$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - qx(i,k) - kq(i,k)d4(i,k) \geq 0; \quad i = 1,\ldots, n \quad k = 1,\ldots, K$$

$$-qx(i,k) + Q(i,k)d4(i,k) \geq 0; \; i = 1,\ldots, n \quad k = 1,\ldots, K$$

$$qw, \; qb \text{ en } qx \geq 0$$

$$0 \leq d1 - d4 \leq 1$$

De structuur van dit gewone lineaire programmeringsprobleem is zodanig dat wanneer de optimale waarden voor qw(i,j) en qb(i,j) bepaald zijn de optimale waarden voor alle andere variabelen ook bekend zijn. Zodoende is het mogelijk om probleem (6.3) te reduceren tot een probleem in qw(i,j) en qb(i,j). Dit kan op de volgende wijze:

- De coëfficiënt A(i) van d1(i) in de doelstellingsfunctie is positief. Omdat het hier een minimaliseringsprobleem betreft, zal d1(i) zo klein mogelijk zijn. Volgens (6.3) moet d1(i) aan de volgende beperkingen voldoen:

$$d1(i) \leq \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\}/rq(i,0) \tag{6.4}$$

$$d1(i) \geq \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\}/QA \tag{6.5}$$

Omdat $rq(i,0) \leq QA$ geldt:

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\}/rq(i,0) \geq \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\}/QA \quad (6.6)$$

Uit (6.6) volgt dat de minimale waarde voor d1(i) welke aan beperking (6.5) voldoet ook aan beperking (6.4) voldoet. Voor de minimimale waarde voor d1(i) geldt dus:

$$d1(i) = \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\}/QA \quad (6.7)$$

- Op dezelfde wijze kan worden afgeleid dat voor d2(i) en d3(i,s) geldt:

$$d2(i) = \sum_{j=1}^{m} \{qw(i,j)\}/QA \quad (6.8)$$

$$d3(i,s) = \sum_{j=1}^{m} \{coef(i,s,j)qb(i,j)\}/QA \quad (6.9)$$

- De coëfficient H(i,k) van qx(i,k) in de doelstellingsfunctie is negatief, qx(i,k) zal dus zo groot mogelijk zijn. Volgens (6.3) moet qx(i,k) aan de volgende beperkingen voldoen:

$$qx(i,k) \leq \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - kq(i,k)d4(i,k) \quad (6.10)$$

$$qx(i,k) \leq Q(i,k)d4(i,k) \leq \{QA - kq(i,k)\}d4(i,k) \quad (6.11)$$

Uit (6.10) en (6.11) kan worden afgeleid dat voor de optimale waarden voor qx(i,k) en d4(i,k) geldt:

$$qx(i,k) = \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\}Q(i,k)/QA \quad (6.12)$$

$$d4(i,k) = \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\}/QA \quad (6.13)$$

Substitutie van de vergelijkingen (6.7) t/m (6.9), (6.12) en (6.13) in het oorspronkelijke probleem (6.3) levert een gereduceerd probleem in qw(i,j) en qb(i,j) op, waarbij een groot aantal beperkingen overbodig wordt. Het gereduceerde probleem kan geschreven worden volgens (6.14):

$$\min \sum_{j=1}^{m} \{psi1(i,j)qw(i,j) + psi2(i,j)qb(i,j)\}$$

$$ST \sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} \geq a(j);\ j = 1,\ldots, m \qquad (6.14)$$

$$qw,\ qb \geq 0$$

$$\text{met } psi1(i,j) = F(i,j) + [A(i) + B(i) + \sum_{k=1}^{K} \{H(i,k)Q(i,k)\}]/QA$$

$$psi2(i,j) = G(i,j) + [A(i) + \sum_{k=1}^{K} \{H(i,k)Q(i,k)\}]/QA + \sum_{s=1}^{S} \{E(i,s)coef(i,s,j)/QB(i,s)\}$$

Bepaal de volgende variabelen:

$$nu(i,j) = 0 \text{ wanneer } psi1(i,j) < psi2(i,j)$$
$$= 1 \text{ wanneer } psi1(i,j) \geq psi2(i,j)$$

$$zeta(i,j) = \{1 - nu(i,j)\}psi1(i,j) + nu(i,j)psi2(i,j)$$

$$q(i,j) = \{1 - nu(i,j)\}qw(i,j) + nu(i,j)qb(i,j)$$

$$i(j) = i \text{ met } zeta(i,j) = \min_{i} zeta(i,j)$$

$$del(i,i(j)) = 1 \text{ wanneer } i = i(j), \text{ anders } del\ (i,i(j)) = 0$$

Het minimaliseringsprobleem (6.14) kan met behulp van deze variabelen geschreven worden volgens (6.15):

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} \{zeta(i,j)q(i,j)\}$$

$$ST \sum_{i=1}^{n} q(i,j) \geq a(j);\ j = 1,\ldots, m \qquad (6.15)$$

$$q \geq 0$$

Het is nu eenvoudig in te zien dat in de optimale oplossing de variabelen q(i,j) met de kleinste coëfficiënt zeta(i,j) gelijk zullen zijn aan a(j) en de overige q(i,j) gelijk zullen zijn aan nul, oftewel

$$q(i,j) = del(i,i(j))a(j) \qquad (6.16)$$

Dit leidt tot de volgende optimale oplossing voor het oorspronkelijke primale probleem (6.3):

$$qw(i,j) = \{1 - nu(i,j)\}del(i,i(j))a(j) \tag{6.17}$$

$$qb(i,j) = nu(i,j)del(i,i(j))a(j) \tag{6.18}$$

$$d1(i) = \sum_{j=1}^{m} \{del(i,i(j)a(j)\}/QA \tag{6.19}$$

$$d2(i) = \sum_{j=1}^{m} [\{1 - nu(i,j)\}del(i,i(j)a(j)]/QA \tag{6.20}$$

$$d3(i,s) = \sum_{j=1}^{m} \{coef(i,s,j)nu(i,j)del(i,i(j))a(j)\}/QA \tag{6.21}$$

$$d4(i,k) = \sum_{j=1}^{m} \{del(i,i(j))a(j)\}/QA \tag{6.22}$$

$$qx(i,k) = \sum_{j=1}^{m} \{del(i,i(j))a(j)Q(i,k)\}/QA \tag{6.23}$$

met als minimale kosten:

$$\begin{aligned} TCG = & \sum_{i=1}^{n} \sum_{j=1}^{m} [a(j)del(i,i(j)\{A(i)/QA + \{1 - nu(i,j)\}B(i)/QA \\ & + F(i,j)\}] + \sum_{i=1}^{n} \sum_{j=1}^{m} [a(j)del(i,i(j))nu(i,j) + \\ & + \sum_{s=1}^{S} \{coef(i,s,j)E(i,s)/QB(i,s) + G(i,j)\}] + \\ & + \sum_{i=1}^{n} \sum_{j=1}^{m} [a(j)del(i,i(j)). \sum_{k=1}^{K} \{H(i,k)Q(i,k)/QA\}] \end{aligned} \tag{6.24}$$

De op deze wijze berekende minimale kosten van het continu veronderstelde minimaliseringsprobleem (6.3) zijn een ondergrens voor de minimale kosten van het gemengd geheeltallige probleem (3.34) t/m (3.41) en de optimale oplossing (6.17) t/m (6.24) is het startpunt voor de berekening de toelaatbare dummievector, die zo dicht mogelijk bij de optimale oplossing ligt. Deze berekening verloopt volgens de volgende stappen:

1. Stel dh1 - dh4 gelijk aan d1 - d4 volgens de vergelijkingen (6.19) t/m (6.22).

Stel d1(i) = 1 als dh1(i) > 0
d1(i) = 0 als dh1(i) = 0; i = 1,..., n
gam(i) = 0; i = 1,..., n

2. Controleer of aan de beperking (6.25) voldaan wordt voor alle i met d1(i) = 1

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} \geq rq(i,0) \qquad (6.25)$$

Zo ja, ga naar stap 3; zo niet, stel dan:

$$gam(i) = rq(i,0) - \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} \qquad (6.26)$$

en voeg gam(i) toe aan alle positieve qw(i,j) en qb(i,j)

3. Bereken de overige dummievariabelen:

d2(i) = 1 als dh2(i) > 0
= 0 als dh2(i) = 0

d3(i,s) = 1 als dh3(i,s) > 0
= 0 als dh3(i,s) = 0

d4(i,k) = 1 als qx(i,k) + gam(i) ≥ kq(i,k)
= 0 als qx(i,k) + gam(i) < kq(i,k)

6.8 Bepalen van de oplossing van het primale probleem, de subroutines PRISOL en RESIM

Het primale probleem (6.1) moet worden opgelost voor gegeven waarden van de dummievariabelen, die in de eerste iteratie van het oplossingsalgorithme berekend zijn in de subroutine ANSOL en in de volgende iteraties in de subroutine ZEON. Hierbij wordt gebruik gemaakt van de subroutines PRISOL (PRImal SOLution) en RESIM (REvised SIMplex method). Listings van de subroutines PRISOL en RESIM en een verklaring van de betekenis van de in de subroutines gebruikte variabelen zijn opgenomen in de bijlagen XIII en XIV.

De subroutine PRISOL bestaat uit de volgende stappen:

1. Reduceren van het primale probleem.

2. Bepalen van een startbasis en een inverse van de coëfficiëntenmatrix van de startbasis.

3. De subroutine RESIM aanroepen om het gereduceerde primale probleem met behulp van de revised simplex methode op te lossen.

4. De oplossing van het gereduceerde primale probleem omzetten in de oplossing van het volledige primale probleem.

1. Reduceren van het primale probleem

Wanneer de gegeven waarden van de dummievariabelen in het primale probleem (6.1) gesubstitueerd worden, kan het probleem geschreven worden volgens (6.27):

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} \{F(i,j)qw(i,j) + G(i,j)qb(i,j)\} + \sum_{i=1}^{n} \sum_{k=1}^{K} \{H(i,k)qx(i,k)\}$$

$$+ \text{res}'(d) \qquad (6.27)$$

$$\text{ST} \sum_{i=1}^{n} \{qw(i,j) + qb(i,j)\} - YS(0,j) = a(j);\ j = 1,\ldots, m$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - YS(1,i,1) = rq(i,0)d1(i);\ i = 1,\ldots n$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - YS(1,i,2) = QA.d1(i); \qquad (*)$$

$$i = 1,\ldots, n$$

$$\sum_{j=1}^{m} \{qw(i,j)\} + YS(2,i) = QA.d2(i);\ i = 1,\ldots, n \qquad (*)$$

$$\sum_{j=1}^{m} \{coef(i,s,j)qb(i,j)\} + YS(3,i,s) = QB(i,s)d3(i,s); \qquad (*)$$

$$i = 1,\ldots, n$$
$$s = 1,\ldots, S$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - qx(i,k) - YS(4,i,k) = kq(i,k)d4(i,k);$$

$$i = 1,\ldots, n$$
$$k = 1,\ldots, K$$

$$qx(i,k) + YS(5,i,k) = Q(i,k)d4(i,k);\ i = 1,\ldots, n \qquad (*)$$
$$k = 1,\ldots, K$$

met $qw(i,j)$, $qb(i,j)$, $qx(i,k)$ en $YS \geq 0$

$d1(i)$, $d2(i)$, $d3(i,s)$ en $d4(i,k)$: 0-1 variabelen

$\text{res}'(d)$: een functie van de dummievariabelen

Dit primale probleem kan alsvolgt vereenvoudigd worden:

- De met een (*) gemarkeerde beperkingen zijn in het kostenmodel geïntroduceerd om de waarden van de dummievariabelen te genereren en kunnen dus vervallen.
- Alle variabelen die corresponderen met de dummievariabelen d1(i) = 0 kunnen vervallen. Stel IO is de verzameling indices i van alle d1(i) met de waarde één oftewel IO = {i | d1(i) = 1}. Het probleem kan nu gereduceerd worden tot de beperkingen waarvoor geldt i ϵ IO.
- Alle (overgebleven) variabelen die corresponderen met de dummievariabelen d4(i,k) = 0 kunnen vervallen.

$$\text{Stel} \quad K(i) = \sum_{k=1}^{K} \{d4(i,k)\}$$

Voor de overgebleven variabelen qx(i,k) geldt nu i ϵ IO en k $\leq$ K(i).

- De dummievariabelen corresponderen met de vaste kosten, dus de minimalisering blijft beperkt tot de variabele kosten. Wanneer de waarden van zowel qw(i,j) als qb(i,j) positief kunnen zijn kan de waarde van de qw(i,j) of qb(i,j) met de grootste coëfficiënt in de doelstellingsfunctie gelijk aan nul gesteld worden.

Stel:

eta(i,j) = 0 wanneer zowel qw(i,j) als qb(i,j) gelijk aan nul moeten zijn
eta(i,j) = 1 wanneer alleen qw(i,j) positief kan zijn of qw(i,j) en qb(i,j) positief kunnen zijn en F(i,j) $\leq$ G(i,j)
eta(i,j) = 2 wanneer alleen qb(i,j) positief kan zijn of qw(i,j) en qb(i,j) positief kunnen zijn en F(i,j) > G(i,j)

$$q(i,j) = eta(i,j).\{2 - eta(i,j)\}.qw(i,j) + (eta(i,j)/2).(eta(i,j) - 1).qb(i,j)$$

$$psi'(i,j) = eta(i,j).\{2 - eta(i,j)\}.F(i,j) + (eta(i,j)/2).(eta(i,j) - 1).G(i,j); \; eta(i,j) > 0$$

psi'(i,j) = LARGE, met LARGE een groot getal; eta(i,j) = 0

- Maak gebruik van de specifieke eigenschappen van de volgende spline functie beperkingen:

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - qx(i,k) - YS(4,i,k) = kq(i,k); \qquad (6.28)$$

$$i = 1,\ldots, n$$
$$k = 1,\ldots, K$$

Omdat de coëfficiënt van qx(i,k) negatief is, zal qx(i,k) zo groot mogelijk zijn. Wanneer nu geldt:

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} \geq kq(i,k) \qquad (6.29)$$

dan kan YS(4,i,k) gelijk aan nul gesteld worden en geldt voor qx(i,k):

$$qx(i,k) = \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - kq(i,k) \qquad (6.30)$$

Omdat rq(i,0) < kq(i,1) < ... < kq(i,K) kan vergelijking (6.30) onder de voorwaarde (6.29) in de doelstellingsfunctie gesubstitueerd worden voor k = 1,..., K(i).

De beperkingen

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - qx(i,k) - YS(4,i,k) = kq(i,k); \quad i = 1,\ldots, n \quad k \leq K(i)$$

$$qx(i,k) + YS(5,i,k) = Q(i,k); \quad i = 1,\ldots, n \quad k \leq K(i)$$

kunnen nu vervangen worden door:

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - h'(i) = kq(i,K(i)); \quad K(i) > 0 \qquad (6.31)$$

met h'(i): een spelingsvariabele

Als gevolg van de substitutie van (6.30) in de doelstellingsfunctie moeten zowel psi'(i,j) als res'(d) aangepast worden:

$$psi(i,j) = psi'(i,j) + \sum_{k=1}^{K} \{d4(i,k).H(i,k)\}$$

$$res(d) = res'(d) - \sum_{k=1}^{K} \{d4(i,k).H(i,k).kq(i,k)\}$$

Omdat rq(i,k) < kq(i,K(i)) voor K(i) > 0 geldt:

$$YS(1,i,1) = \sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - rq(i,0) > 0$$

zodat ook deze beperking kan vervallen:

Stel:
$$\begin{aligned} ha(i) &= kq(i,K(i)) && \text{als } K(i) > 0 \\ &= rq(i,0) && \text{als } K(i) = 0 \\ h(i) &= h'(i) && \text{als } K(i) > 0 \\ &= YS(1,i,1) && \text{als } K(i) = 0 \end{aligned}$$

De beperkingen

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j\} - YS(1,i,1) = rq(i,0)$$

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j\} - h'(i) = kq(i,K(i)) \quad \text{voor } K(i) > 0$$

kunnen nu vervangen worden door:

$$\sum_{j=1}^{m} \{qw(i,j) + qb(i,j)\} - h(i) = ha(i) \tag{6.32}$$

Na het toepassen van de hiervoor beschreven vereenvoudigingen kan het primale probleem alsvolgt geschreven worden:

$$\min \sum_{i=1}^{n} \sum_{j=m}^{m} \{psi(i,j).q(i,j)\} + res(d) \tag{6.33}$$

$$ST \quad \sum_{i=1}^{n} \{q(i,j)\} - YS(0,j) = a(j);\ j = 1,\ldots, m$$

$$\sum_{j=1}^{m} \{q(i,j)\} - h(i) = ha(i);\ i \in IO$$

met $q(i,j)$, $YS(0,j)$ en $h(i) \geq 0$

Het aantal beperkingen van het primale probleem is nu teruggebracht van $\{m+n(3+S+2K)\}$ naar $(m+N)$ met

$$N = \sum_{i=1}^{n} \{d1(i)\}$$

2. Bepalen van een startbasis en een inverse van de coëfficiëneenmatrix van de startbasis.

Een goede startbasis voor de revised simplex methode moet aan de volgende voorwaarden voldoen:

- De variabelen van de startbasis moeten zo dicht mogelijk bij de optimale oplossing liggen. Dit wil zeggen dat de variabelen, waarvan verwacht kan worden dat ze in de optimale oplossing voorkomen, als basisvariabelen gekozen worden.
- De startbasis moet toelaatbaar zijn.
- De coëfficiëntenmatrix van de startbasis moet bij voorkeur eenvoudig te inverteren zijn.

De volgende basisvariabelen worden gekozen:

- De q(i,j) die corresponderen met de laagste kosten per vraagpunt. Deze basisvariabelen corresponderen met de beperking:

$$\sum_{i=1}^{n} \{q(i,j)\} - YS(0,j) = a(j);\ j = 1,\ldots, m - 1 \qquad (6.34)$$

Deze variabelen worden gekozen omdat het primale probleem een minimaliseringprobleem is en alle coëfficiënten positief zijn.

- De variabelen die corresponderen met de beperkingen:

$$\sum_{j=1}^{m} \{q(i,j)\} - h(i) = ha(i);\ i = 1,\ldots, N - 1 \qquad (6.35)$$

Dit zijn:

$$h(i) \quad \text{wanneer} \sum_{j=1}^{m} \{q(i,j)\} \geq ha(i)$$

of

$$q(i,m) \quad \text{wanneer} \sum_{j=1}^{m} \{q(i,j)\} < ha(i)$$

- Nu zijn (m-1)+(N-1) basisvariabelen gekozen. De laatste twee basisvariabelen corresponderen met de beperkingen:

$$\sum_{i=1}^{N-1} \{q(i,m)\} + q(N,m) - YS(0,m) = a(m) \qquad (6.36)$$

$$\sum_{j=1}^{m-1} \{q(N,m)\} + q(N,m) - h(N) = ha(N) \qquad (6.37)$$

Als basisvariabelen worden gekozen:

$$\text{Wanneer } ha(N) - \sum_{i=1}^{N} \{q(i,m)\} \le a(m) - \sum_{j=1}^{m} \{q(N,j)\} \implies h(N)$$

$$\text{en wanneer } ha(N) - \sum_{i=1}^{N} \{q(i,m)\} \le 0 \implies YS(0,m)$$

$$\text{anders} \implies q(N,m)$$

$$\text{Wanneer } ha(N) - \sum_{i=1}^{N} \{q(i,m)\} > a(m) - \sum_{j=1}^{m} \{q(N,j)\} \implies YS(0,m)$$

$$\text{en wanneer } a(m) - \sum_{j=1}^{m} \{q(N,j)\} \le 0 \implies h(n)$$

$$\text{anders} \implies q(N,m)$$

De coëfficiëntenmatrix van deze basisvariabelen wordt weergegeven in fig. 25.

	m-1	1	N-1	1	
	I	0	0	0	m-1
	0	a	{a(i)+1}/2	c	1
AB =	d(i,j)	0	a(i,k)	0	N-1
	d(N,j)	b	{a(i)+1}/2	d	1

Fig. 25 Coëfficiëntenmatrix van de basisvariabelen

met:

q(N,m) en YS(0,m) basisvariabelen: a, b = 1, c = -1, d = 0
q(N,m) en h(N) basisvariabelen: a, b = 1, c = 0, d = -1
YS(0,M) en h(N) basisvariabelen: a, b = 1, c = -1, d = 0
q(i,j) > 0 : d(i,j) = 1
q(i,j) ≤ 0 : d(i,j) = 0
i ongelijk k : a(i,k) = 0
q(i,m) basisvariabele : a(i,i) = a(i) en a(i) = 1
h(i) basisvariabele : a(i,i) = a(i) en a(i) = -1

De inverse van de matrix AB is weergeven in de fig. 26 t/m 28.

$AB^{-1}=$

m-1	1	N-1	1	
I	0	0	0	m-1
-d(n,j)	0	0	1	1
-d(i,j)a(i)	0	a(i,k)	0	N-1
-b(j)-d(N,j)	-1	{a(i)+1}/2	1	1

Fig. 26 Inverse matrix AB^{-1} met q(N,m) en YS(0,m) als basisvariabelen

Met $b(j) = \sum_{i=1}^{N-1} \{d(i,j)(a(i) + 1)/2\}$.

$AB^{-1} =$

m-1	1	N-1	1	
I	0	0	0	m-1
b(j))	0	-{a(i)+1}/2	0	1
-d(i,j)a(i)	0	a(i,k)	0	N-1
b(j)+d(N,j)	1	-{a(i)+1}/2	-1	1

Fig. 27 Inverse matrix AB^{-1} met q(N,m) en h(N) als basisvariabelen

$AB^{-1} =$

m-1	1	N-1	1	
I	0	0	0	m-1
-b(j)	-1	(a(i)+1)/2	0	1
-d(i,j)a(i)	0	a(i,k)	0	N-1
d(N,j)	0	-(a(i)+1)/2	-1	1

Fig. 28 Inverse matrix AB^{-1} met YS(0,M) en h(N) als basisvariabelen

3. Het gereduceerde primale probleem met behulp van de revised simplex methode oplossen.

Zie [12] voor een beschrijving van de revised simplex methode.

4. De oplossing van het gereduceerde primale probleem omzetten in de oplossing van het volledige primale probleem.

Stel het gereduceerde primale probleem heeft de volgende optimale oplossing: q(i,j), h(i) en YS(0,j).

De optimale oplossing van het volledige primale probleem kan nu verkregen worden door de optimale oplossing van het gereduceerde primale probleem te substitueren in de volgende vergelijkingen:

$$qw(i,j) = eta(i,j)(2 - eta(i,j))q(i,j)$$

$$qb(i,j) = (eta(i,j)(eta(i,j) - 1)/2)q(i,j) = q(i,j) - qw(i,j)$$

$$qx(i,k) = (\sum_{j=1}^{m} (q(i,j)) - kq(i,k))d4(i,k)$$

$$YS(0,j) = YS(0,j)$$

$$YS(1,i,1) = d1(i)(\sum_{j=1}^{m} (q(i,j)) - rq(0,i))$$

$$YS(1,i,2) = d1(i)[QA - \sum_{j=1}^{m} (q(i,j))]$$

$$YS(2,i) = d2(i)[QA - \sum_{j=1}^{m} (qw(i,j))]$$

$$YS(3,i,s) = d3(i,s)[QB(i,s) - \sum_{j=1}^{m} \{coef(i,s,j)qb(i,j)\}]$$

$$YS(4,i,k) = \{1 - d4(i,k)\} \sum_{j=1}^{m} \{q(i,j)\}$$

$$YS(5,i,k) = d4(i,k)[Q(i,k) + kq(i,k) - \sum_{j=1}^{m} \{q(i,j)\}]$$

6.9 Transformeren van de oplossing van het primale probleem naar de oplossing van het duale probleem, de subroutines TRANS en CORRECT

In de subroutine TRANS (**TRANS**formation of the primal problem) wordt de oplossing van het primale probleem, zoals bepaald in de subroutines PRISOL en RESIM, getransformeerd naar de oplossing van het duale probleem. De subroutine CORRECT (**CORRECT** dual variables) wordt aangeroepen in het geval dat de transformatie leidt tot negatieve waarden voor de duale variabelen; CORRECT corrigeert deze waarden. Listings van de subroutines TRANS en CORRECT en een verklaring van de betekenis van de in de subroutines gebruikte variabelen zijn opgenomen in de bijlagen XV en XVI.

In (6.2) is het duale probleem geformuleerd. Zoals reeds vermeld is, corresponderen de duale variabelen hierbij met de primale variabelen volgens tabel 5.

primaal	duaal	primaal	duaal
qw(i,j)	XS(1,i,j)	YS(1,i,2)	u(1,i,2)
qb(i,j)	XS(2,i,j)	YS(2,i)	u(2,i)
qx(i,k)	XS(3,i,k)	YS(3,i,s)	u(3,i,s)
YS(0,j)	u(0,j)	YS(4,i,k)	u(4,i,k)
YS(1,i,1)	u(1,i,1)	YS(5,i,k)	u(5,i,k)

Tabel 5 Relatie tussen de primale en de duale variabelen

Van ieder paar primale - duale variabele is slechts één variabele basisvariabele. Dit wil zeggen dat wanneer de primale basisvariabelen bekend zijn, de duale basisvariabelen eveneens bekend zijn. Om de duale basisvariabelen te bepalen moet een stelsel vergelijkingen worden opgelost, waarbij het aantal vergelijkingen gelijk is aan het aantal variabelen. In deze paragraaf wordt het stelsel vergelijkingen beschreven en een methode om dit stelsel op te lossen.

De duale variabelen die corresponderen met de primale basisvariabelen zijn gelijk aan nul:

- u(1,i,2), u(2,i) en u(3,i,s) wanneer de corresponderende dummievariabelen d1(i), d2(i) of d3(i,s) gelijk aan één zijn. De reden hiervoor is dat de corresponderende primale variabelen de spelingsvariabelen zijn in de beperkingen die de waarde van deze dummievariabelen bepalen en dat deze spelingsvariabelen basisvariabelen zijn.
- u(1,i,1), u(4,i,k) en XS(3,i,k) wanneer de corresponderende dummievariabele d1(i) gelijk aan nul is. De corresponderende primale variabelen YS(1,i,1), YS(4,i,k) en qx(i,k) kunnen beschouwd worden als basisvariabelen.
- Wanneer d1(i) = 1:
  . XS(3,i,k) en u(4,i,k) voor k > K(i). De corresponderende primale variabelen qx(i,k), die gemaximaliseerd worden en YS(4,i,k) kunnen beschouwd worden als basisvariabelen.
  . XS(3,i,k) en u(5,i,k) voor k < K(i)
  . u(1,i,1) voor k ≥ K(i). Omdat rq(i,0) < kq(i,1) zal YS(1,i,1) basisvariabele zijn.

De duale variabelen die basisvariabele moeten zijn:

- XS(1,i,j) corresponderend met qw(i,j) = 0
- XS(2,i,j) corresponderend met qb(i,j) = 0
- u(1,i,2) corresponderend met d1(i) = 0
- u(2,i) corresponderend met d2(i) = 0
- u(3,i,s) corresponderend met d3(i,s) = 0
- u(5,i,k); k > K(i)
- u(4,i,k); k ≤ K(i)
- u(0,j) corresponderend met YS(0,j) = 0
- u(1,i,1); d1(i) = 1 en K(i) = 0
- XS(3,i,K(i)); K(i) > 1

Wanneer de bekende niet-basisvariabelen worden weggelaten kunnen de beperkingen van het duale optimaliseringsprobleem alsvolgt geschreven worden:

- voor d1(i) = 0:

$$u(0,j) - u(1,i,2) - u(2,i) + XS(1,i,j) = F(i,j); \quad j = 1,\ldots, m \tag{6.38}$$

$$u(0,j) - u(1,i,2) - \sum_{s=1}^{S} \{coef(i,s,j).u(3,i,s)\} + XS(2,i,j) = G(i,j); \quad j = 1,\ldots, m \tag{6.39}$$

$$u(5,i,k) = -H(i,k);\ k = 1,\ldots, K \tag{6.40}$$

- voor d1(i) = 1 en K(i) = 0:

$$u(0,j) + u(1,i,1) - u(2,i) + XS(1,i,j) = F(i,j);\ j = 1,\ldots, m \tag{6.41}$$

$$u(0,j) - u(1,i,1) - \sum_{s=1}^{S} \{coef(i,s,j).u(3,i,s)\} + XS(2,i,j) = G(i,j);$$
$$j = 1,\ldots, m \quad (6.42)$$

$$u(5,i,k) = -H(i,k);\ k = 1,\ldots, K \quad (6.43)$$

- voor d1(i) = 1 en K(i) ≥ 0:

$$u(0,j) - u(2,i) + \sum_{k=1}^{K} \{u(4,i,k)\} + XS(1,i,j) = F(i,j); \quad (6.44)$$
$$j = 1,\ldots, m$$

$$u(0,j) - \sum_{s=1}^{S} \{coef(i,s,j).u(3,i,s)\} + \sum_{k=1}^{K} \{u(4,i,k)\} +$$

$$+ XS(2,i,j) = G(i,j);\ j = 1,\ldots, m \quad (6.45)$$

$$u(4,i,k) = -H(i,k);\ k = 1,\ldots, K(i)-1 \quad (6.46)$$

$$u(4,i,k) - XS(3,i,k) = -H(i,k);\ k = K(i) \quad (6.47)$$

$$u(5,i,k) = -H(i,k);\ k = K(i)+1,\ldots, K \quad (6.48)$$

In deze vergelijkingen zijn om wille van de notatie een aantal bekende niet-basisvariabelen opgenomen.

Uit (6.44) t/m (6.48) kan worden afgeleid, dat wanneer K(i) > 0:

$$\begin{aligned} u(4,i,k) &= -H(i,k); && k = 1,\ldots, K(i)-1 \\ &= XS(3,i,k) - H(i,k); && k = K(i) \\ &= 0; && k = K(i)+1,\ldots, K \end{aligned}$$

Uit (6.38) t/m (6.43) kan worden afgeleid, dat wanneer K(i) = 0:

$$u(4,i,k) = 0;\ k = 1,\ldots, K$$

Stel $F'(i,j) = F(i,j) + \sum_{k=1}^{K} \{d4(i,k).H(i,k)\}$

en $G'(i,j) = G(i,j) + \sum_{k=1}^{K} \{d4(i,k).H(i,k)\}$

dan kunnen de vergelijkingen (6.44) t/m (6.48) vervangen worden door:

$$u(0,j) - u(2,i) + XS(1,i,j) + XS(3,i,K(i)) = F'(i,j)$$

$$u(0,j) - \sum_{s=1}^{S} \{coef(i,s,j).u(3,i,s)\} + XS(2,i,j) + XS(3,i,K(i)) = F'(i,j)$$

$$u(5,i,k) = -H(i,k);\ k = K(i)+1,\ldots,\ K$$

Ten aanzien van het stelsel vergelijkingen kan gesteld worden:

- Dat in iedere vergelijking één basisvariabele bepaald wordt.
- Dat de basisvariabelen XS(1,i,j), XS(2,i,j) en u(5,i,k) in slechts één vergelijking voorkomen. Deze variabelen worden bepaald als functie van de overige basisvariabelen: u(0,j), u(2,i), u(3,i,s), XS(3,i,k(i), u(1,i,1) en u(1,i,2).
- Dat de waarde van de basisvariabelen u(2,i), u(3,i,s) en u(1,i,2) arbitrair gekozen moet worden omdat deze variabelen een coëfficiënt ter groote nul in de doelstellingsfunctie hebben en de optimale oplossing niet beïnvloeden. Een methode voor deze arbitraire keuze wordt later aangegeven.
- Dat de waarde van de overgebleven basisvariabelen bepaald moet worden in de vergelijkingen die corresponderen met XS(1,i,j) of XS(2,i,j) gelijk aan nul (oftewel qw(i,j) of qb(i,j) basisvariabele).

Stel:

$$\begin{aligned}
z(i) &= u(1,i,1) && \text{als } K(i) = 0\\
&= XS(3,i,K(i) && \text{als } K(i) \geq 1\\
psi(i,j) &= F'(i,j) && \text{als } qw(i,j) > 0\\
&= G'(i,j) && \text{als } qb(i,j) > 0\\
NNV(j) &= \text{het aantal } i \text{ met } \{i \mid qw(i,j) > 0 \text{ of } qb(i,j) > 0\}\\
NV(j,jh) &= \{i \mid qw(i,j) > 0 \text{ of } qb(i,j) > 0\};\ jh = 1,\ldots,NNV(j)
\end{aligned}$$

Het duale probleem kan nu gereduceerd worden tot het bepalen van de variabelen u(0,j) en z(i).

$$u(0,j) + z(i) = psi(i,j);\ j = 1,\ldots,\ m \qquad i \in NV(j,jh) \qquad jh = 1,\ldots,\ NNV(j) \tag{6.49}$$

Immers, indien qw(i,j) > 0 geldt d2(i) = 1 waardoor u(2,i) = 0 en indien qb(i,j) > 0 geldt d3(i,s) = 1 als coef(i,s,j) = 1 waardoor u(3,i,s) = 0

Voor het oplossen van de vergelijking (6.49) wordt de volgende methode gehanteerd:

1. Stel z(i) = 0 en GI(i) = 1 wanneer d1(i) = 0 of d1(i) = 1 en z(i) correspondeert met een positieve primale basisvariabele.

   GI(i) is een vector die aangeeft of z(i) gewaardeerd is (GI(i) = 1) of niet (GI(i) = 0).

2. Bepaal de waarde van de variabelen u(0,j) met j $\in$ J1(jh)

u(0,j) = 0 wanneer YS(0,j) > 0

u(0,j) = psi(i,j) - z(i) wanneer er een index i bestaat waarvoor geldt dat GI(i) = 1 en XS(1,i,j) = 0 of XS(2,i,j) = 0.

J1(jh) is de verzameling van indices j van de u(0,j) die nog niet gewaardeerd zijn. In de eerste iteratiestap is J1(jh) gelijk aan j = 1, m, NJ = m en jh = 1, NJ; NJ is het aantal indices van de verzameling J1(jh).

Bepaal voor i ϵ NV(j,jh) (jh = 1, NNV(j)) en de j waarvan u(0,j) gewaardeerd zijn:
z(i) = psi(i,j) - u(0,j) en GI(i) = 1

Bepaal de nieuwe verzameling J1(jh) en de waarde van NJ.

3. Indien NJ = 0 is de oplossing gevonden. Ga naar stap 6.

   Indien in stap 2 tenminste één z(i) gewaardeerd is, herhaal dan stap 2.

4. Indien NJ gelijk is aan het aantal vergelijkingen dat nog niet gebruikt is om een variabele u(0,j) of z(i) te waarderen dan geldt:

u(0,j) = psi(i,j) en z(i) = 0 voor i ϵ NV(j,jh); j = 1,..., NNV(j)

   De oplossing is nu gevonden. Ga naar stap 6.

5. Er zijn een aantal vergelijkingen (6.49) overgebleven met j ϵ J1(jh) en GI(i) = 0. Dit betekent dat er meer vergelijkingen zijn dan variabelen u(0,j) en z(i). De oplossing kan alleen gevonden worden door tenminste één variabele arbitrair te bepalen.

   Stel:

   I2 is de verzameling indices van z(i) die gewaardeerd zijn nadat één z(i) arbitrair bepaald is.

   J2 is de verzameling indices van u(0,j) die gewaardeerd zijn nadat één u(0,j) arbitrair bepaald is.

   z(i2) = 0; i2 ϵ I2 met i2 = NV(j2,1) en j2 = J1(1)

   Nu geldt:

   u(0,j2) = psi(i2,j2) met j2 ϵ J2

   z(i) = psi(i,j2) - u(0,j2) met i ϵ I2 en i ϵ NV(j2,jh)
   Indien er een z(i) = a < 0 voorkomt voor i ϵ I2 dan moet er een correctie worden uitgevoerd:

$z(i) = z(i) - a$ voor alle $i \in I2$

$u(0,j) = u(0,j) + a$ voor alle $j \in J2$

Indien er een $u(0,j) = b < 0$ voorkomt voor $j \in J2$ dan moet er eveneens gecorrigeerd worden:

$u(0,j) = u(0,j) - b$ voor alle $j \in J2$

$z(i) = z(i) + b$ voor alle $i \in I2$

Keer vervolgens terug naar stap 2.

6. De oplossing van het stelsel vergelijkingen (6.49) is gevonden. Nu moeten de waarden van $u(1,i,2)$, $u(2,i)$ en $u(3,i,s)$ berekend worden:

$$d1(i) = 0 \quad : u(1,i,2) = \max_j \{u(0,j) - \min\{F(i,j), G(i,j)\}\}$$

$$d1(i) = 1 \quad : u(1,i,2) = 0$$

$$d2(i) = 0 \quad : u(2,i) = \max_j \{u(0,j) - F'(i,j)\}$$

$$d2(i) = 1 \quad : u(2,i) = 0$$

$$d3(i,s) = 0 : u(3,i,s) = \max_j \{coef(i,s,j).(u(0,j) - G'(i,j))\}$$

$$d3(i,s) = 1 : u(3,i,s) = 0$$

De oplossing van het stelsel (6.49) en de variabelen $u(1,i,2)$, $u(2,i)$ en $u(3,i,s)$ moeten getransformeerd worden naar de oplossing van het duale probleem:

$$XS(1,i,j) = F'(i,j) - u(0,j) - z(i) + u(1,i,2) + u(i,2)$$

$$XS(2,i,j) = G'(i,j) - u(0,j) - z(i) + u(1,i,2) + \sum_{s=1}^{S} \{coef(i,s,j).u(3,i,s)\}$$

$$XS(3,i,k) = z(i) \quad \text{als } k = K(i)$$
$$= 0 \quad \text{als } k \neq K(i)$$

$$u(1,i,1) = z(i) \quad \text{als } K(i) = 1$$
$$= 0 \quad \text{als } K(i) \neq 1$$

$$u(4,i,k) = XS(3,i,k) - H(i,k) \quad \text{als } d4(i,k) = 1$$
$$= 0 \quad \text{als } d4(i,k) = 0$$

$$u(5,i,k) = -H(i,k) \quad \text{als } d4(i,k) = 0$$
$$= 0 \quad \text{als } d4(i,k) = 1$$

6.10 Bepaling van de dummievariabelen, de subroutine ZEON

In de subroutine ZEON wordt het minimum van het duale probleem bepaald t.a.v. de dummievariabelen voor gegeven waarden van de continue variabelen. De continue variabelen worden bepaald in de subroutines PRISOL, RESIM en TRANS en hebben de vorm van een constante be(t) en coëfficiënten gam(t,j) van de dummievariabelen x1(j). Per iteratie van het hoofdprogramma MAIN wordt een vergelijking met als bekenden be(t) en gam(t,j) en als onbekenden de dummievariabelen x1(j) aan het in ZEON op te lossen minimaliseringsprobleem toegevoegd.

Het middels ZEON op te lossen minimaliseringsprobleem heeft de volgende vorm:

$$\min_{x1} z \tag{6.50}$$

$$\begin{array}{lll} \text{ST} & z \geq & al(1) = be(1) + \sum_{j=1}^{ND} gam(1,j)x1(j) \\ & & \vdots \\ & z \geq & al(T) = be(T) + \sum_{j=1}^{ND} gam(T,j)x1(j) \end{array}$$

met be(t) : constante in iteratie t van het oplossingsalgorithme [f]
gam(t,j): coëfficiënt van de dummievariabele x1(j) in iteratie t van het oplossingsalgorithme [f]
ND : aantal dummievariabelen [-]
T : aantal doorlopen iteraties van het oplossingsalgorithme [-]
x1(j) : dummievariabele [-]

Zoals reeds opgemerkt, is de oplossingsmethode voor het bepalen van de dummievariabelen aangepast t.o.v. de oorspronkelijke versie welke beschreven wordt in [12]. In de aangepaste versie van ZEON is de onderlinge afhankelijkheid van de dummievariabelen verwerkt. In deze paragraaf wordt de aangepaste versie van ZEON besproken. In bijlage XVII is een listing van de subroutine ZEON opgenomen en wordt de betekenis van de in de subroutine gebruikte variabelen verklaard.

De subroutine ZEON bestaat uit de volgende stappen:

0. Bepalen van startwaarden voor een aantal variabelen.

1. Waarderen van de combinaties van x1(j) met G(j) = 2.

2. Bepalen van de pivotrij.

3. Bepalen van de pivotvariabele.

4. Waarderen van de pivotvariabele.

5. Reeds gewaardeerde pivotvariabelen mogelijk herwaarderen.

6. Bepalen van de optimale oplossing z.

7. Bepalen van de optimale oplossing z voor één vergelijking.

8. Controleren en eventueel aanpassen van de waarden van de dummievariabelen x1(j).

9. Opnieuw berekenen van de optimale oplossing z.

0. Bepalen van startwaarden voor een aantal variabelen

- M : nummer van de iteratiestap in ZEON; startwaarde M = 0.

- a^* : minimale waarde voor z in de voorgaande iteratie van ZEON; startwaarde $a^* = \min_t \{be(t)\}$.

- G(j): vector die aangeeft of x1(j) gewaardeerd is;

  startwaarden:

  G(j) = 1 en x1(j) = 0: x1(j) is niet gewaardeerd, gam(t,j) > 0 voor minstens één t of gam(t,j) = 0 voor alle t.
  G(j) = 0 en x1(j) = 1: comb(j,0) = 0 en gam (t,j) ≤ 0 voor alle t met gam(t,j) < 0 voor tenminste één t.
  G(j) = 2 en x1(j) = 0: comb(j,0) > 0 en gam(t,j) ≤ 0 voor alle t met gam(t,j) < 0 voor tenminste één t; x1(j) wordt mogelijk gewaardeerd in stap 1.

- FI(j): de minimale waarde van z voor x1(j) = 1 - x1(j) als G(j) = 0; startwaarden FI(j) = 0.

- $x^*(j)$: waarden van de dummievariabelen x1(j) waarvoor a^* bereikt wordt; startwaarden $x^*(j) = 0$.

- GN : aantal j waarvoor geldt G(j) = 2.

1. Waarderen van de combinaties van x1(j) met G(j) = 2

- Indien GN = 0 zijn er geen combinaties van dummievariabelen die in stap 1 mogelijk gewaardeerd worden: naar stap 2.

- NMUT: aantal combinaties van dummievariabelen dat in stap 1 gewaardeerd is; startwaarde NMUT = 0.

- Voor $\{j' | G(j') = 2\}$ met $\sum_{j \in J^*} ((1 - x1(j))gam(t,j)) \leq 0$ voor alle t

  met J^*: j' en alle j die in de combinatie van j' voorkomen

geldt voor j': $G(j') = 0$ en $GN = GN + 1$
$x1(j') = 1$ en $NMUT = NMUT + 1$

en voor $j \epsilon J^*$: $GN = GN - 1$ als $G(j) = 2$
$x1(j) = 1$ en $G(j) = 0$.

- Indien $NMUT > 0$, d.w.z. er zijn combinaties gewaardeerd, dan kan $\sum_{j \epsilon J^*} \{(1 - x1(j))gam(t,j)\}$ welke eerst > 0 was

  voor één of meer $t \leq 0$ geworden zijn voor alle t door de waardering van de x1(j): herhaal stap 1.

2. Bepalen van de pivotrij

- Indien TZ = 1, d.w.z. er is maar één vergelijking: naar stap 7.

- Als $G(j) = 2$: $G(j) = 1$.

- MMAX:aantal dummievariabelen dat nog gewaardeerd moet worden;

  $$\text{startwaarde } MMAX = \sum_{j=1}^{ND} \{G(j)\}.$$

- $M = M + 1$: het nummer van de iteratiestap binnen ZEON wordt met één verhoogd.

- $$al(t) = be(t) + \sum_{j=1}^{ND} \{gam(t,j)x1(j)\}.$$

- Als $a^* > \max_t \{al(t)\}$: $a^* = \max_t \{al(t)\}$.

- $x^*(j)$: de x1(j) waarvoor a^* bereikt wordt.

- t1: de t waarvoor a^* bereikt wordt; het nummer van de pivotrij.

3. Bepalen van de pivotvariabele

- Indien $M > MMAX$ d.w.z. allle dummievariabelen zijn gewaardeerd: naar stap 5.

- $$MIN = \min_{j'} \sum_{j \epsilon J^*} \{G(j)gam(t1,j)\}.$$

  met J^*: j' en de j die in de combinatie van j' voorkomen

- j1: de j' waarvoor MIN bereikt wordt

- IN(M) = j1; in stap 5 wordt nagegaan of z verminderd kan worden door de reeds gewaardeerde pivotvariabelen opnieuw te waarderen. Per iteratiestap wordt daarom geregistreerd welke de pivotvariabele in die stap is.

- Als MIN $\geq$ 0 d.w.z. vermindering van al(t1) is niet mogelijk: naar stap 5.

4. Waarderen van de pivotvariabele

- G(j1) = 0; de pivotvariabele x1(j1) wordt gewaardeerd.

- $am(0) = \max_{t} \{al(t) + \sum_{j \epsilon J^-} \{G(j)gam(t,j)\}\}$

  met J^-: de j waarvoor geldt $\{G(j)gam(t,j)\} < 0$

- $am(1) = am(0) + \max_{t} \{gam(t,j1) + \sum_{j \epsilon J^+} \{G(j)gam(t,j)\}\}$

  met J^+: de j die in de combinatie van j1 voorkomen en waarvoor geldt $\{G(j)gam(t,j)\} \geq 0$

- am(0) < am(1): l1 = 0, anders l1 = 1.

- x1(j1) = l1, FI(j1) = am(1-l1)

- De pivotvariabele x1(j1) is nu gewaardeerd. Indien x1(j1) = 1 moet gecontroleerd worden of er geen x1(j) uit de combinatie van x1(j1) voorkomen, die reeds gewaardeerd zijn en de waarde nul hebben. Zo ja, dan moet ook x1(j1) de waarde nul krijgen:

  Als comb(j1,0) > 0 voor $j \epsilon J^*$ en G(j) + x1(j) = 0

  met J^*: de j die in de combinatie van j1 voorkomen

  dan geldt: x1(j1) = 0 en FI(j1) = as + 1

- Als x1(j1) = 1 en comb(j1,0) > 0 dan moeten de dummievariabelen die in de combinatie van x1(j1) voorkomen en nog niet gewaardeerd zijn eveneens de waarde één krijgen:

  comb(j1,0).l1 > 0 en voor $j \epsilon J^*$ geldt G(j) = 1: G(j) = 0 en x1(j) = 1 voor $j \epsilon J^*$

- Indien door de waardering van de x1(j) uit de combinatie van x1(j1) geldt am(l1) > a^* dan draagt de waardering van de pivotvariabele niet bij aan een verlaging van het minimum:

  G(j1) = 1, x1(j1)= 0 en naar stap 5.

- Analoog aan de bepaling van IN(M) moet ook de waardering van de dummievariabelen die in de combinatie van x1(j1) voorkomen, geregistreerd worden:

  MUT(M,jh) = j voor $j \epsilon J^*$

  met J^*: de j die in de combinatie van j1 voorkomen.

- De volgende pivotvariabele moet nu gewaardeerd worden: naar stap 2.

5. Reeds gewaardeerde pivotvariabelen mogelijk herwaarderen

- M = M - 1; ga één stap terug.

- Als M = 0: ga naar stap 6.

- j1 = IN(M); de pivotvariabele uit de betreffende stap wordt opgeroepen.

- Als FI(j1) $\geq a^*$:

  G(j1) = 1, x1(j1) = 0.
  MUT(M,0) = 0: d.w.z. er zijn in de betreffende iteratiestap geen x1(j) uit de combinatie van x1(j1) gewaardeerd: herhaal stap 5.
  MUT(M,0) > 0: G(j) = 1 en x1(j) = 0 voor j ϵ J^* en MUT(M,jh) = 0 met J^*: de j die in de combinatie van j1 voorkomen.

  Stel MUT(M,0) = 0 en herhaal stap 5.

- Als FI(j1) < a^*:

  x1(j1) = 1 - x1(j1), FI(j1) = a^* + 1

  Als x1(j1) = 0:

  MUT(M,0) = 0: naar stap 2.
  MUT(M,0) > 0: G(j) = 1 en x1(j) = 0 voor j ϵ MUT(M,jh), MUT(M,0) = 0, naar stap 2.

  Als x1(j1) = 1:

  MUT(M,0) = 0

  Als comb(j1,0) = 0: naar stap 2

  Als comb(j1,0) > 0:

  Voor j ϵ J^*:

  als G(j) = 1 : G(j) = 0, x1(j) = 1, en voeg de gewaardeerde variabelen aan MUT(M,jh) toe.
  als x1(j) = 0: FI(j1) = a^* + 1, G(j1) = 1 en x1(j1) = 0, G(j) = 1 en x1(j) = 0 voor j ϵ MUT(M,jh), herhaal stap 5.

  Keer vervolgens terug naar stap 2

6. Bepalen van de optimale oplossing z

- z = a^*, x1(j) = x^*(j) voor j = 1, ND.

- Ga naar stap 8.

7. Bepalen van de optimale oplossing z voor één vergelijking

$$- z = be(1) + \sum_{j=1}^{ND} x1(j)gam(1,j)$$

8. controleren en eventueel aanpassen van de waarden van de dummievariabelen x1(j)

- Controleer of er tenminste één dummievariabele x1(j) die correspondeert met het openen van de winputten de waarde één gekregen heeft. Zo niet: x1(1) = 1, x1(2) = 1 en ga naar stap 9.

- Controleer of er transport vanuit de geopende winputten is voorzien:

  . Als de x1(j) die behoren bij de geopende winputten en corresponderen met transport per vrachtwagen gelijk aan één zijn, is aan deze controle voldaan.
  . Zo niet, als dan de x1(j) die behoren bij de geopende winputten en die corresponderen met de leidingstukken die direct aan de vraagpunten verbonden zijn gelijk aan één zijn, is aan deze controle voldaan.
  . Indien niet aan deze controle voldaan wordt: x1(j) = 1 voor de x1(j) die behoren bij de geopende winputten en corresponderen met transport per vrachtwagen.

9. Opnieuw berekenen van de optimale oplossing z

$$- z = \max_{t} al(t) + \sum_{j=1}^{ND} (x1(j)gam(t,j))$$

7. Ontwikkelen en testen van het programma

7.1 Inleiding

Het in het voorgaande hoofdstuk beschreven computerprogramma, dat is ontwikkeld om het minimaliseringsprobleem op te lossen, is de laatste versie van het programma. Het programma heeft gedurende het onderzoek een zekere ontwikkeling doorgemaakt.

In dit hoofdstuk zulllen niet alle aanpassingen, die in de loop der tijd hebben plaatsgevonden, worden besproken. Besproken worden de ontwikkeling van de oorspronkelijke versie, waarvan de theorie beschreven is in [12] en welke deels gebaseerd is op een door het ICW ter beschikking gestelde broncode en de ontwikkeling van de laatste versie.

Ontwikkeling van de oorspronkelijke versie van het programma

Zoals reeds gemeld, is de oorspronkelijke versie gebaseerd op de theorie uit [12]. Aan de hand van deze theorie en van de beschikbare broncode, welke niet volledig was en nooit gedraaid heeft, is een programmastructuur ontworpen. Op basis van deze programmastructuur zijn een hoofdprogramma en een aantal subroutines geschreven in FORTRAN 77. Deze subroutines zijn allen getest en voor de subroutines RECALCO en TRANS zijn bovendien nog tests uitgevoerd van specifieke onderdelen.

De volgende tests zijn uitgevoerd:

- Test van de berekening van de spline functie variabelen
- Test van de subroutine RECALCO
- Test van de subroutine ANSOL
- Test van de subroutine RESIM
- Test van de subroutine PRISOL
- Test van de berekening van de duale variabelen
- Test van de subroutine TRANS
- Test van de subroutine ZEON

In paragraaf 7.2 t/m 7.9 worden deze subroutinetests besproken.

Bij het draaien van het volledige programma in zijn oorspronkelijke versie bleek dat het niet tot een oplossing leidde. In paragraaf 7.10 is getracht te verklaren wat hiervan de reden is. Op basis van deze verklaring is het programma aangepast; deze aanpassingen worden eveneens besproken in paragraaf 7.10.

Ontwikkeling van de laatste versie van het programma

Het oorspronkelijke hoofdprogramma MAIN en de oorspronkelijke subroutines zijn aangepast. Omdat dit m.u.v. de aanpassing van de subroutine ZEON niet tot principiële veranderingen heeft geleid, zijn ze niet opnieuw getest. Wel getest is het hulpprogramma MODIFY, dat deel uitmaakt van de laatste versie van het volledige programma (zie paragraaf 6.4). De test van MODIFY wordt beschreven in paragraaf 7.11.

De subroutine ZEON is ingrijpend aangepast. De nieuwe versie van ZEON (zie paragraaf 6.10) is echter niet separaat getest. Door de onderlinge afhankelijkheid van de dummievariabelen bleek het zeer moeilijk om een testprobleem te formuleren. Daarom is besloten de subroutine ZEON te testen door het volledige programma te draaien. Hiermee is een betrouwbare test van ZEON te verwachten, omdat alle andere subroutines immers getest zijn.

In paragraaf 7.12 worden twee runs van het volledige programma besproken, waarvan er één wel tot een oplossing leidde en de ander niet. Getracht wordt te verklaren wat de reden is van het niet goed functioneren van het programma.

7.2 Test van de berekening van de spline functie variabelen

Met het programma SPLINE is de berekening getest van de variabelen, waarmee de lineaire spline functie gedefinieerd wordt. Zie voor de spline functie theorie [13].

In het kostenmodel is bij de spline functie benadering uitgegaan van de relatieve afwijking ρ. Vandaar dat ook deze test met de relatieve afwijking ρ wordt uitgevoerd.

De invoergegevens voor deze test zijn afkomstig uit [13] waarin twee voorbeelden van een spline functie berekening zijn opgenomen. De uitkomsten van de test komen overeen met die van het rekenvoorbeeld met de relatieve afwijking ρ.

In bijlage XVIII is de listing van het programma SPLINE opgenomen en wordt de betekenis van de gebruikte variabelen verklaard. Daarnaast is in bijlage XVIII de in- en uitvoer van het rekenvoorbeeld opgenomen.

7.3 Test van de subroutine RECALCO

De subroutine RECALCO is getest met behulp van het programma TESTRECA. In dit programma wordt de subroutine RECALCO aangeroepen en worden de in de subroutine ingelezen invoergegevens en de berekende coëfficiënten uitgevoerd.

In de subroutine RECALCO worden de invoergegevens van het minimaliseringsprobleem ingelezen en worden de coëfficiënten van het kostenmodel berekend. De berekening van de spline functie variabelen maakt hier deel van uit. Omdat deze berekening reeds getest is (zie de voorgaande paragraaf), wordt ze in deze test niet gecontroleerd. Bij de berekening van de spline functie variabelen is in deze test eveneens uitgegaan van de relatieve afwijking ρ.

De invoer van deze test wordt gevormd door de gegevens van een deel van het ICW-proefgebied, bestaande uit de winputten 4 en 18 en de vraagpunten 8, 30, 39, 40 en 41. De invoer komt overeen met die van het programma INPUT (zie paragraaf 5.3 en bijlage VI). Het aantal leidingtrajecten is hierbij handmatig

teruggebracht van 86 tot 25 door die trajecten, die geen deel uitmaken van het deelnetwerk dat de 2 winputten met de 5 vraagpunten verbindt, te laten vervallen. De berekende coëfficiënten zijn steekproefsgewijs door middel van een handberekening gecontroleerd.

In bijlage XIX zijn de listings van het programma TESTRECA en de subroutine RECALCO opgenomen en wordt de betekenis van de gebruikte variabelen verklaard. Daarnaast zijn in bijlage XV de in- en uitvoer van de subroutinetest alsmede de handmatige controleberekening opgenomen.

7.4 Test van de subroutine ANSOL

De subroutine ANSOL is getest met behulp van het programma TESTANSOL. In dit programma worden de subroutines RECALCO en ANSOL aangeroepen en worden de in ANSOL berekende dummievariabelen en het berekende minimum uitgevoerd.

In de subroutine ANSOL worden de startwaarden voor de dummievariabelen berekend. De invoer is identiek aan die van de test van de subroutine RECALCO (bijlage XIX). De tussenuitkomsten van ANSOL en de berekende variabelen zijn steekproefsgewijs door middel van een handberekening gecontroleerd.

De subroutine RECALCO is identiek aan die van de test van RECALCO (bijlage XIX) m.u.v. de communicatie met de subroutine ANSOL. In bijlage XX is dit aangegeven.

In bijlage XX zijn de listings van het programma TESTANSOL en de subroutine ANSOL opgenomen en wordt de betekenis van de gebruikte variabelen verklaard. Daarnaast zijn in bijlage XX tussenuitkomsten van ANSOL en de berekende variabelen opgenomen, alsmede de handmatige controleberekening.

7.5 Test van de subroutine RESIM

De subroutine RESIM is getest met behulp van het programma TESTRESIM. In het programma TESTRESIM worden de invoergegevens van de testproblemen ingelezen en uitgevoerd, wordt de subroutine RESIM aangeroepen en worden de berekende waarden uitgevoerd.

In totaal zijn drie kleine problemen getest waarvan de uitkomsten handmatig gecontroleerd zijn. De testproblemen zijn zodanig gekozen, dat de inverse van de coëfficiëntenmatrix van de startbasisvariabelen voor deze problemen eenvoudig te berekenen is (bijvoorbeeld de eenheidsmatrix). Dit is overeenkomstig de benadering in het complete programma, waar de coëfficiëntenmatrix van de startbasisvariabelen de eenheidsmatrix is.

De listings van het programma TESTRESIM en de subroutine RESIM en een verklaring van de betekenis van de gebruikte variabelen zijn opgenomen in bijlage XXI. Daarnaast is in bijlage XXI de in- en uitvoer opgenomen van de volgende drie testproblemen:

Probleem 1

$$\min z = -3x_1 - 5x_2$$

$$\begin{aligned} \text{ST} \quad x_1 \qquad\qquad &\leq 4 \\ x_2 &\leq 6 \\ 3x_1 + 2x_2 &\leq 18 \\ x_{1-2} &\geq 0 \end{aligned}$$

Onder toevoeging van de spelingsvariabelen kan dit probleem alsvolgt geschreven worden:

$$\min z = -3x_1 - 5x_2$$

$$\begin{aligned} \text{ST} \quad x_1 \qquad + x_3 \qquad\qquad &= 4 \\ x_2 \qquad + x_4 \qquad &= 6 \\ 3x_1 + 2x_2 \qquad\qquad + x_5 &= 18 \\ x_{1-5} &\geq 0 \end{aligned}$$

De spelingsvariabelen x_3, x_4 en x_5 worden gekozen als startbasisvariabelen. De coëfficiëntenmatrix van deze basisvariabelen en haar inverse zijn gelijk aan de eenheidsmatrix.

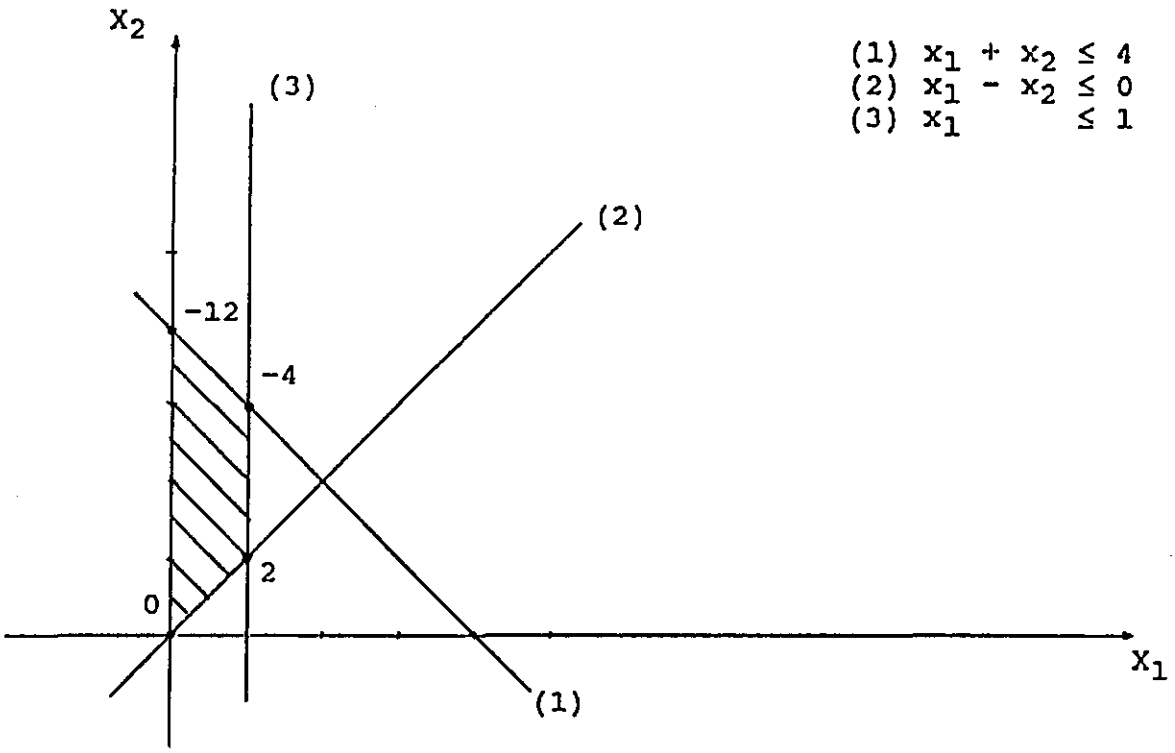
Probleem 1 is als voorbeeld opgenomen in [12], echter als maximaliseringsprobleem. In deze test is het probleem omgezet in een minimaliseringsprobleem door de coëfficiënten van de objectfunctie een tegengesteld teken te geven. De uitkomsten van deze test komen overeen met die van het voorbeeld van [12]; het in het voorbeeld gevonden maximum ($z = 36$) is in deze test echter een minimum ($z = -36$).

Probleem 2

$$\min z = 5x_1 - 3x_2$$

$$\begin{aligned} \text{ST} \quad x_1 + x_2 &\leq 4 \\ x_1 - x_2 &\leq 0 \\ x_1 &\leq 1 \\ x_{1-2} &\geq 0 \end{aligned}$$

De spelingsvariabelen worden op dezelfde wijze aan het probleem toegevoegd als in probleem 1. Ook hier zijn de spelingsvariabelen x_3, x_4 en x_5 gekozen als startbasisvariabelen en zijn zowel de coëfficiëntenmatrix van deze basisvariabelen als haar inverse gelijk aan de eenheidsmatrix. De oplossing van dit probleem is grafisch weergegeven in fig. 29 en de uitkomsten van deze test komen hiermee overeen.



(1) $x_1 + x_2 \leq 4$
(2) $x_1 - x_2 \leq 0$
(3) $x_1 \leq 1$

Fig. 29 Grafische oplosssing van testprobleem 2

<u>Probleem 3</u>

Dit probleem verschilt van probleem 2 doordat de beperkingen tegengesteld zijn:

$$\min z = 5x_1 - 3x_2$$

$$\begin{aligned} \text{ST} \quad & x_1 + x_2 \geq 4 \\ & x_1 - x_2 \geq 0 \\ & x_1 \geq 1 \\ & x_{1-2} \geq 0 \end{aligned}$$

Onder toevoeging van de spelingsvariabelen kan dit probleem geschreven worden als:

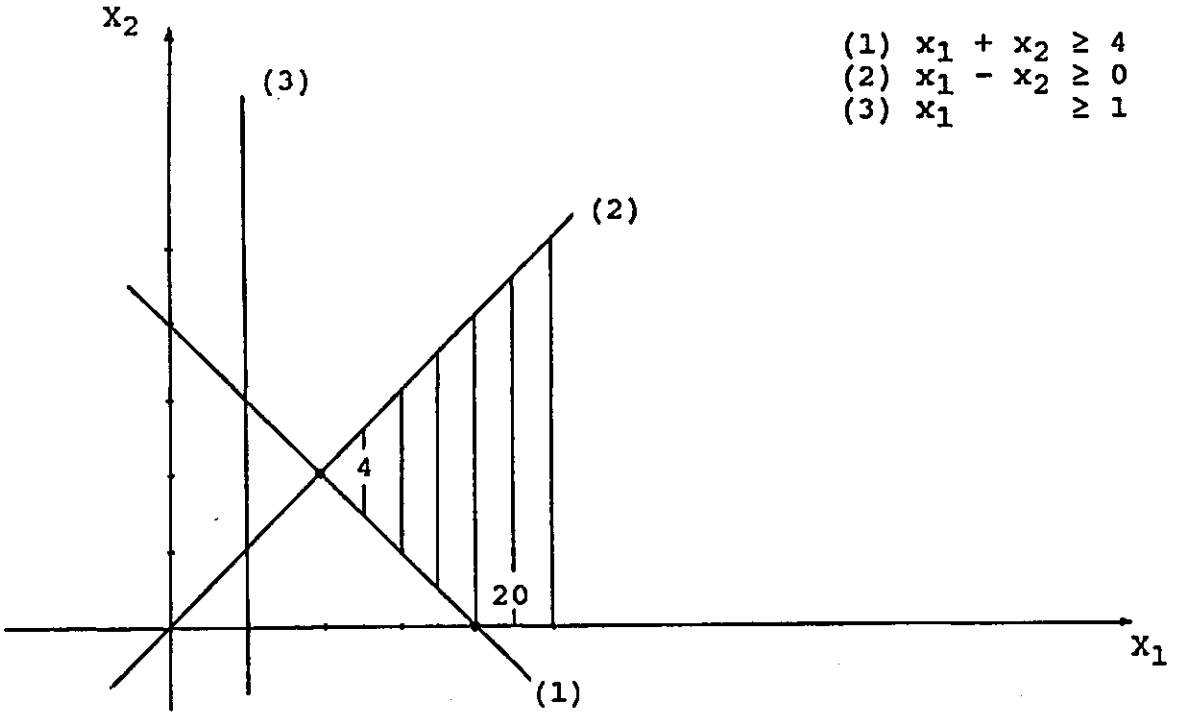
$$\min z = 5x_1 - 3x_2$$

$$\begin{aligned} \text{ST} \quad & x_1 + x_2 - x_3 = 4 \\ & x_1 - x_2 - x_4 = 0 \\ & x_1 - x_5 = 1 \\ & x_{1-5} \geq 0 \end{aligned}$$

De variabelen x_1, x_4 en x_5 worden gekozen als startbasisvariabelen. Zowel de coëfficiëntenmatrix van deze basisvariabelen als haar inverse zijn dan gelijk aan:

$$B = B^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

De oplossing van dit probleem is grafisch weergegeven in fig. 30 en de uitkomsten van deze test komen hiermee overeen.



(1) $x_1 + x_2 \geq 4$
(2) $x_1 - x_2 \geq 0$
(3) $x_1 \geq 1$

Fig. 30 Grafische oplossing van testprobleem 3

7.6 Test van de subroutine PRISOL

De subroutine PRISOL is getest met behulp van het programma TESTPRISOL. In dit programma worden de subroutines RECALCO, ANSOL en PRISOL aangeroepen en worden de in PRISOL berekende continue variabelen en het berekende minimum uitgevoerd.

In de subroutine PRISOL worden de continue variabelen berekend voor gegeven waarden van de dummievariabelen, die berekend worden in de subroutine ANSOL. De invoer is identiek aan die van de test van de subroutine RECALCO (bijlage XIX). Tevens worden tussenuitkomsten uitgevoerd. Deze tussenuitkomsten en de berekende variabelen zijn steekproefsgewijs door middel van een handberekening gecontroleerd.

De subroutine RECALCO is identiek aan die van de test van RECALCO (bijlage XIX) m.u.v. de communicatie met de subroutines ANSOL en PRISOL. In bijlage XXII is dit aangegeven.

De subroutine ANSOL is identiek aan die van de test van ANSOL (bijlage XX) zonder het uitvoeren van de tussenuitkomsten en m.u.v. de communicatie met de subroutines RECALCO en PRISOL. In bijlage XXII is dit aangegeven.

Vanuit de subroutine PRISOL wordt de subroutine RESIM aangeroepen. Deze subroutine is identiek aan die van de test van RESIM (bijlage XXI) m.u.v. enige statements. In bijlage XXII is aangegeven wat de verschillen zijn.

In bijlage XXII zijn de listings van het programma TESTPRISOL en de subroutine PRISOL opgenomen en wordt de betekenis van de gebruikte variabelen verklaard. Tevens zijn in deze bijlage de tussenuitkomsten uit PRISOL en de berekende variabelen opgenomen alsmede de handmatige controleberekening.

7.7 Test van de berekening van de duale variabelen

De berekening van de duale variabelen, wat een belangrijk onderdeel is van de subroutine TRANS (zie paragraaf 6.9) is getest middels het programma TESTDUAL. In dit programma wordt, zoals in de subroutine TRANS, zonodig de subroutine CORRECT aangeroepen.

De test wordt uitgevoerd met het volgende stelsel vergelijkingen:

u0(1) + z(1) = psi(1,1) = 1.0

u0(2) + z(1) = psi(1,2) = 1.0

u0(3) + z(1) = psi(1,3) = 1.0

u0(4) + z(2) = psi(2,4) = 2.0

u0(4) + z(3) = psi(3,4) = 1.0

u0(5) + z(2) = psi(2,5) = 1.0

In matrixvorm kan dit stelsel alsvolgt geschreven worden:

$$\left[\begin{array}{ccccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} u0(1) \\ u0(2) \\ u0(3) \\ u0(4) \\ u0(5) \\ z(1) \\ z(2) \\ z(3) \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.0 \\ 1.0 \\ 2.0 \\ 1.0 \\ 1.0 \end{bmatrix}$$

De test is gecontroleerd door alle stappen handmatig te volgen; hierbij is gebleken dat de methode ter oplossing van het stelsel vergelijkingen goed functioneerde. Van deze controle is geen verslag gedaan.

In bijlage XXIII zijn de listings van het programma TESTDUAL en de subroutine CORRECT opgenomen en wordt de betekenis van de gebruikte variabelen verklaard. Daarnaast zijn in bijlage XXIII de in- en uitvoer van deze test opgenomen.

7.8 Test van de subroutine TRANS

De subroutine TRANS is getest met behulp van het programma TESTTRANS. In dit programma worden de subroutines RECALCO, ANSOL, PRISOL en TRANS aangeroepen en worden de in TRANS berekende duale variabelen en het berekende maximum uitgevoerd.

In de subroutine TRANS worden de duale variabelen berekend op basis van de oplossing van het primale probleem (opgelost met PRISOL). De invoer is identiek aan die van de test van de subroutine RECALCO (bijlage XIX). Tevens worden tussenuitkomsten uitgevoerd. De tussenuitkomsten en de berekende variabelen zijn steekproefsgewijs door middel van een handberekening gecontroleerd.

De subroutine RECALCO is identiek aan die van de test van RECALCO (bijlage XIX) m.u.v. de communicatie met de subroutines ANSOL, PRISOL en TRANS. In bijlage XXIV is dit aangegeven.

De subroutine ANSOL is identiek aan die van de test van ANSOL (bijlage XX) zonder het uitvoeren van de tussenuitkomsten en m.u.v. de communicatie met de subroutines RECALCO, PRISOL en TRANS. In bijlage XXIV is dit aangegeven.

De subroutine PRISOL is identiek aan die van de test van PRISOL (bijlage XXII) zonder het uitvoeren van de tussenuitkomsten en m.u.v. de communicatie met de subroutines RECALCO, ANSOL, PRISOL en TRANS. In bijlage XXIV is dit aangegeven.

Vanuit de subroutine PRISOL wordt de subroutine RESIM aangeroepen. Deze subroutine is identiek aan die van de test van RESIM (bijlage XXI) m.u.v. enige statements. In bijlage XXIV is aangegeven wat de verschillen zijn.

De subroutine CORRECT wordt zonodig aangeroepen vanuit de subroutine TRANS. Zij is identiek aan die van de test van de berekening van de duale variabelen (bijlage XXIII) m.u.v. de communicatie met de subroutine TRANS. In bijlage XXIV is dit aangegeven.

In bijlage XXIV zijn de listings van het programma TESTTRANS en de subroutine TRANS opgenomen en wordt de betekenis van de gebruikte variabelen verklaard. Tevens zijn in deze bijlage de tussenuitkomsten uit TRANS en de berekende variabelen opgenomen alsmede de handmatige controleberekening.

7.9 Test van de subroutine ZEON

De subroutine ZEON is getest met behulp van het programma TESTZEON. Het betreft hier de test van de oorspronkelijke subroutine ZEON, gebaseerd op de zoekmethode voor de dummievariabelen zoals beschreven in [12].

In het programma TESTZEON worden de invoergegevens van het testprobleem ingelezen, wordt de subroutine ZEON aangeroepen en worden de berekende waarden uitgevoerd.

In de subroutine ZEON worden de dummievariabelen bepaald volgens de genoemde zoekmethode.

Als testprobleem is het voorbeeld uit [12] gebruikt, aangevuld met de dummievariabelen x_6 en x_7. Van deze 0-1 variabelen is eenvoudig vooraf in te zien, wat hun waarden zullen zijn ($x_6 = 0$ en $x_7 = 1$). Het testprobleem heeft de volgende vorm:

$$\min_{x_i} z$$

$$\begin{aligned}
\text{ST } z &\geq 10 - 8x_1 + 15x_2 + 6x_3 + 2x_4 - 11x_5 + x_6 - x_7 \\
z &\geq 13 + 2x_1 - 10x_2 + 3x_3 - x_4 - 6x_5 + x_6 \\
z &\geq 9 - 5x_1 - 5x_2 + 8x_3 + 2x_4 + x_6 - x_7 \\
z &\geq 15 - 3x_1 - 6x_2 + 11x_3 - 3x_4 - 4x_5 + x_6 \\
z &\geq 11 + 8x_1 + x_2 - 10x_3 + x_4 + x_5 + x_6 - x_7
\end{aligned}$$

De uitkomsten van de test komen overeen met die van het voorbeeld ($z = 12 - 1 = 11$; -1 als gevolg van de toegevoegde dummievariabele $x_7 = 1$).

In bijlage XXV zijn de listings van het programma TESTZEON en de subroutine ZEON opgenomen en wordt de betekenis van de gebruikte variabelen verklaard. Daarnaast is in bijlage XXV de in- en uitvoer van het testprobleem opgenomen.

7.10 Oorspronkelijke en laatste versie van het programma

Het volledige programma is gedraaid in de oorspronkelijke versie. Hierbij is gebruik gemaakt van de oorspronkelijke versie van het hoofdprogramma MAIN. De listing van MAIN alsmede de betekenis van de in MAIN gebruikte variabelen en het parameterbestand PROGRAM.PAR zijn opgenomen in bijlage XXVI.

Het programma is gedraaid voor hetzelfde deelprobleem als bij de programmarun van MPSX. Dit deelprobleem bestaat uit de potentiële winputten 4 en 18 en de vraagpunten 8, 30, 39, 40 en 41. In fig. 21 is het deelgebied van dit deelprobleem gemarkeerd als

onderdeel van het volledige proefgebied. De invoergegevens van het deelprobleem zijn opgenomen in bijlage VI.

Uit de uitvoer, die is opgenomen in bijlage XXVI, blijkt dat het programma in de derde iteratiestap wordt afgebroken omdat er een niet-toelaatbare startbasis voor de revised simplex methode (subroutine RESIM) gegenereerd wordt.

De oorzaak van het afbreken van het programma moet gezocht worden in de subroutine ZEON. Wanneer in ZEON een oplossing voor de dummievariabelen gevonden wordt die geen reële betekenis heeft en dus ontoelaatbaar* is, dan wordt het primale probleem op basis van deze ontoelaatbare oplossing gereduceerd en ook de startbasis voor de revised simplex methode wordt op basis van deze oplossing bepaald. Hierdoor zal deze startbasis niet toelaatbaar zijn.

De noodzaak om de onderlinge afhankelijkheid van de dummievariabelen op te nemen in de subroutine ZEON is hiermee duidelijk.

Voor het programma heeft dit de volgende consequenties gehad:

1. Het hulpprogramma MODIFY is toegevoegd; dit hulpprogramma wordt uitgebreid beschreven in paragraaf 6.4.

2. De nummering van de dummievariabelen is aangepast; in de oorspronkelijke versie van het programma werden de dummievariabelen weergegeven door de variabelen d1(i), d2(i), d3(i,s) en d4(i,k). Doordat er tussen deze variabelen onderlinge verbanden gelegd zijn, bestond de noodzaak ze door één variabele weer te geven: de variabele x1(jd) met jd = 1,..., ND. De relatie tussen de oude en de nieuwe nummering is hierbij de volgende:

$$d1(i) \Rightarrow x1(jd) \text{ met } jd = (i-1)*N + \sum_{i=1}^{i-1} si(i) + (i-1)*K + 1$$

$$d2(i) \Rightarrow x1(jd) \text{ met } jd = (i-1)*N + \sum_{i=1}^{i-1} si(i) + (i-1)*K + 2$$

$$d3(i,s) \Rightarrow x1(jd) \text{ met } jd = (i-1)*N + \sum_{i=1}^{i-1} si(i) + (i-1)*K + 2 + s$$

$$d4(i,k) \Rightarrow x1(jd) \text{ met } jd = (i-1)*N + \sum_{i=1}^{i} si(i) + (i-1)*K + 2 + k$$

---

* Een voorbeeld van een ontoelaatbare oplossing is: d1(i)=0 en d2(i)=1, d.w.z. er is transport van zand per vrachtwagen vanuit winput i, terwijl deze winput niet in produktie genomen is.

7.11 Test van het hulpprogramma MODIFY

Met het programma TESTMODIFY is het hulpprogramma MODIFY getest. In TESTMODIFY worden de invoergegevens van het leidingennet ingelezen en worden de gegevens van het vereenvoudigde leidingennet en de combinatiematrix uitgevoerd.
TESTMODIFY test de vereenvoudiging van het leidingennet en de samenstelling van de combinatiematrix, zoals beschreven in paragraaf 6.4.

Het programma is getest middels een proefgebied bestaande uit de winputten 4 en 18 en de vraagpunten 8, 30, 39, 40 en 41. In tegenstelling tot de eerder beschreven subroutinetests is het leidingennet van het proefgebied niet handmatig vereenvoudigd, dit is immers een van de taken van het hulpprogramma MODIFY.

De uitvoer van MODIFY is handmatig gecontroleerd.

In bijlage XXVII is de listing opgenomen van het programma TESTMODIFY en wordt de betekenis van de gebruikte variabelen verklaard. Daarnaast is de in- en uitvoer opgenomen alsmede de handmatige controle.

7.12 Twee runs van de laatste versie van het programma

De laatste versie van het programma, dat uitgebreid beschreven is in hoofdstuk 6, is gedraaid voor twee testproblemen. De listing van de laatste versie van het programma bestaat uit de listings van het hulpprogramma MODIFY, het hoofdprogramma MAIN en alle subroutines. Deze listings en de betekenis van de gebruikte variabelen zijn opgenomen in de bijlagen IX t/m XVII.

Het programma is gedraaid voor de volgende testproblemen:

1. Het deelprobleem met de winputten 4 en 18 en de vraagpunten 8, 30, 39, 40 en 41

   Dit probleem is identiek aan dat van de programmarun van MPSX en de programmarun met de oorspronkelijke versie van het programma (paragraaf 7.10). Het deelgebied van dit probleem is afgebeeld in fig. 21 (paragraaf 5.3). De invoer voor dit deelprobleem is opgenomen in bijlage XXVIII.

   Uit de uitkomsten voor dit deelprobleem, die weergegeven zijn in bijlage XXVIII, blijkt dat een correcte oplossing verkregen is. De uitkomsten zijn gelijk aan die van de MPSX-run, die bovendien middels een handberekening gecontroleerd is.

2. Het deelprobleem met de winputten 6 en 19 en de vraagpunten 1, 3, 7, 8 en 40

   De invoer voor dit deelprobleem is opgenomen in bijlage XXVIII.

Uit de uitkomsten voor dit deelprobleem, die weergegeven zijn in bijlage XXVIII, blijkt dat er geen oplossing verkregen is. Het programma is wederom afgebroken in de derde iteratiestap omdat er een niet-toelaatbare startbasis voor de revised simplex methode (subroutine RESIM) gegenereerd is.

In paragraaf 7.10 wordt een run besproken van de oorspronkelijke versie van het programma met testprobleem 1. Deze run leidde niet tot een oplossing. Nadat het programma is aangepast is voor hetzelfde deelprobleem wèl een oplossing verkregen. Voor testprobleem 2 wordt geen oplossing verkregen. De conclusie die hieruit getrokken kan worden, is dat de aangepaste zoekprocedure voor de 0-1 variabelen (de subroutine ZEON) soms wèl en soms niet goed functioneert. Immers, voor testprobleem 1 wordt de onderlinge afhankelijkheid van de dummievariabelen goed verwerkt en voor testprobleem 2 niet, hetgeen tot gevolg heeft dat een niet-toelaatbare startbasis voor de revised simplex methode wordt gegenereerd.

De conclusie is dat om het programma naar behoren te laten functioneren de zoekprocedure voor de 0-1 variabelen zodanig moet worden aangepast, dat met de onderlinge afhankelijkheid van de dummievariabelen in alle gevallen rekening wordt gehouden.

Gezien de beschikbare tijd is afgezien van een verdere aanpassing van de subroutine ZEON. Dit zal onderwerp kunnen zijn van verdere studie.

8. Het kostenmodel als beleidsondersteunend instrument

8.1 Inleiding

Zoals in paragraaf 1.1 gesteld is, moet het kostenmodel inzicht geven in de invloed van het aantal, de omvang en de ligging van de zandwinputten op de totale kosten van de ophoogzandvoorziening in een bepaald gebied. Dit inzicht kan dan bijdragen aan het maken van een verantwoorde keuze tussen de diverse potentiële zandwinlocaties.

In het verslag is tot nog toe in hoofdzaak ingegaan op het kostenmodel, zoals dit door het ICW geformuleerd is, en op een drietal oplossingsalgorithmen. De vraag of betreffend kostenmodel inderdaad bijdraagt aan de beleidsvorming t.a.v. de verlening van ontgrondingsvergunningen is nog niet beantwoord; ze komt in dit hoofdstuk aan de orde.

Wil het kostenmodel praktische betekenis hebben als beleidsondersteunend instrument dan moet het aan de volgende vijf voorwaarden voldoen:

1. Het moet in voldoende mate waarheidsgetrouw zijn.

2. Er moeten beleidsalternatieven mee kunnen worden afgewogen.

3. Het moet operationeel zijn, d.w.z. er moet een oplossingsalgorithme beschikbaar zijn waarmee het kostenmodel kan worden doorgerekend.

4. Het moet organisatorisch mogelijk zijn om met het kostenmodel te werken.

5. De baten bij het gebruik van het kostenmodel moeten opwegen tegen de kosten.

Het karakter van het kostenmodel wordt bepaald door de uitgangspunten, die bij het formuleren van het kostenmodel zijn gehanteerd; de keuze voor de modelparameters volgt hierbij uit de uitgangspunten.

In paragraaf 3.2 zijn de belangrijkste uitgangspunten geformuleerd:

- Het tijdaspect blijft in het model buiten beschouwing; de kostenminimalisering wordt uitgevoerd over een bepaalde periode, waarbinnen de vraag naar zand per vraagpunt geraamd wordt.
- De grootte van een zandwinput wordt bepaald door de vraag, met dien verstande dat er per zandwinobject een minimale vraag is, waarbeneden het object niet in produktie wordt genomen.
- Bij de vraag naar zand wordt niet gespecificeerd welke korreldiameter gevraagd wordt; de korreldiameter van het zand heeft wel invloed op de winnings- en transportkosten.
- De winning van het zand vindt plaats middels een drijvende winzuiger.

- Het transport van het zand vindt plaats per vrachtauto of per pijpleiding.
- De volgende kostencategorieën zijn in het model opgenomen:
  . De aankoopkosten van de benodigde terreinen.
  . De aanlegkosten; dit zijn de kosten die gemaakt worden voordat een zandwinput in produktie genomen kan worden.
  . De kosten van de winning en het transport van het zand.

In de paragrafen 8.2 t/m 8.7 wordt aangegeven welke invloed de uitgangspunten en de daaruit volgende modelparameters hebben op de eerste drie voorwaarden waaraan het kostenmodel moet voldoen:

1. Model waarheidsgetrouw

2. Alternatief beleid mogelijk

3. Model operationeel

Zoals uit de hoofstukken 5 en 7 blijkt, is in dit afstudeeronderzoek geen oplossingsalgorithme ter beschikking gekomen waarmee het kostenmodel voor een probleem van enige omvang (zoals het proefgebied) kan worden opgelost. Aan de derde voorwaarde wordt dus niet voldaan. Het moet echter mogelijk zijn om of het huidige oplossingsalgorithme aan te passen of om een alternatief algorithme te ontwikkelen, zodanig dat het kostenmodel operationeel wordt. Bij het aangeven van de invloed die de uitgangspunten en de modelparameters op het operationeel zijn van het kostenmodel hebben, wordt daarom aangegeven wat de invloed is op het beschikbaar komen van een oplossingsalgorithme.

Doordat er geen operationeel oplossingsalgorithme beschikbaar was, heeft er geen kwantitatieve analyse kunnen plaatsvinden van de invloed van de verschillende uitgangspunten en modelparameters. In de paragrafen 8.2 t/m 8.7 wordt daarom alleen ingegaan op de kwalitatieve aspecten.

In paragraaf 8.8 komen de vierde en vijfde voorwaarde aan de orde:

4. Werken met kostenmodel organisatorisch mogelijk

5. Baten bij gebruik kostenmodel wegen op tegen kosten

8.2 Het tijdaspect ontbreekt

Kenmerkend voor transportproblemen in de Operationele Analyse is dat het tijdaspect buiten beschouwing blijft. Dit is ook het geval bij het onderhavige kostenmodel.

Model waarheidsgetrouw

- Het ontbreken van het tijdaspect heeft met name invloed op de waarheidsgetrouwheid van de vraag. Wil met het kostenmodel een zinnige afweging gemaakt kunnen worden tussen de verschillende potentiële zandwinlocaties, dan moet de vraag met enige

nauwkeurigheid worden geschat. Er moet een schatting gemaakt worden van de volgende kenmerkende aspecten van de vraag:
. de periode waarin de vraag per vraagpunt optreedt
. de gevraagde hoeveelheid per vraagpunt
. de locaties van de vraagpunten
Dit brengt met zich mee dat de beschouwde periode niet te kort mag zijn. Om de onzekerheid over de verdeling van de vraag in de tijd zo klein mogelijk te houden zal er een periode gekozen moeten worden van toch minstens enkele jaren.

- Doordat de vraag over een langere periode beschouwd wordt, wordt er geen rekening gehouden met het eventueel optreden van een piekvraag binnen deze periode. Door de piekvraag kan ophoogzand een schaarser artikel worden hetgeen een verhogende invloed kan hebben op de kosten van de terreinen en op de kosten van winning en transport van het ophoogzand.

- Een ander aspect waarmee geen rekening wordt gehouden is een eventuele fluctuatie van loon- en energiekosten binnen de beschouwde periode.

Alternatief beleid mogelijk

- De mogelijkheid bestaat om verschillende vraagscenario's te bestuderen. Hierbij kan gedacht worden aan scenario's waarbij een hoge, een gemiddelde of een lage vraag optreedt en aan scenario's waarbij bepaalde grote projecten, die een grote vraag naar ophoogzand op een bepaalde plaats met zich meebrengen, al dan geen doorgang vinden. De invloed van deze scenario's op de uiteindelijke keuze van de winputten kan bepaald worden, hetgeen een bijdrage kan leveren aan de beleidsvorming.

- Ook voor de prijsfluctuaties van loon- en energiekosten kunnen verschillende scenario's bestudeerd worden. Dat alternatieve beleidsvorming zeer belangrijk is kan worden geïllustreerd aan de hand van het voorstelbare scenario waarbij er binnen de beschouwde periode gelijktijdig sprake is van zowel een piekvraag als van zeer hoge loon- en energiekosten.

Model Operationeel

- Door het buiten beschouwing blijven van het tijdaspect is de kans op het beschikbaar komen van een oplossingsalgorithme groter. Met de introductie van het tijdaspect zullen zowel de vraag, de kostenparameters als de winnings- en transportvariabelen tijdafhankelijk worden. Hierdoor ontstaat een gecompliceerder model waarvoor het vinden van een geschikt oplossingsalgorithme moeilijker zal zijn. Daarnaast is het aannemelijk dat de benodigde rekentijd en het geheugenbeslag bij computerberekening eveneens zullen toenemen.

De conclusie is dat het buiten beschouwing blijven van het tijdaspect aanvaardbaar is, mits de invloed van een eventuele piekvraag en fluctuatie van loon- en energiekosten gering is of in het kostenmodel wordt opgenomen. Hierbij weegt zwaar dat de

kans op het beschikbaar komen van een oplossingsalgorithme door introductie van het tijdaspect aanzienlijk afneemt.

8.3 De grootte van de zandwinput wordt bepaald door de vraag

Model waarheidsgetrouw

- Dat het kostenmodel vraaggestuurd is en niet aanbodgestuurd, is overeenkomstig de werkelijkheid. Het winnen van zand uit een bepaalde locatie komt vrijwel altijd voort uit het optreden van een bepaalde vraag en niet uit initiatief van de eigenaar of beheerder van het betreffende terrein. Vanzelfsprekend beschikt de overheid wel over beleidsinstrumenten om de keuze voor bepaalde zandwinlocaties te beïnvloeden.
- Het hanteren van een ondergrens voor de grootte van de zandwinput* is overeenkomstig de werkelijkheid. Uit het oogpunt van kostenminimalisering is deze ondergrens niet noodzakelijk; het resultaat van de kostenminimalisering zal nooit het in produktie nemen van een kleine zandwinput zijn wanneer de totale kosten hierdoor hoger zullen uitvallen. In de praktijk zal echter geen winput in gebruik worden genomen waarbij na beëindiging van de zandwinning een waterplas overblijft van een zodanige grootte, dat ze geen recratieve of landschappelijke functie kan vervullen, zelfs wanneer dit uit kostenoogpunt aantrekkelijk is.
- Dat in het kostenmodel geen bovengrens aan de grootte van de zandwinput gesteld wordt, is in strijd met de werkelijkheid. In de praktijk zal er vrijwel altijd een bovengrens gehanteerd worden, uit planologische en landschappelijke overwegingen of uit praktische overwegingen zoals bij zandwinputten waarvan de grootte beperkt is door aanwezige bebouwing of infrastructuur.

Alternatief beleid mogelijk

- Dat het kostenmodel vraaggestuurd is, is essentieel voor de mogelijkheid tot het afwegen van beleidsalternatieven. Bij een aanbodgestuurd model zou bij een gegeven vraag alleen nog een afweging van de transportwijze kunnen plaatsvinden.
- De aanwezigheid van een ondergrens voor de grootte van de zandwinput is een belangrijk element bij het afwegen van beleidsalternatieven, terwijl het ontbreken van een bovengrens hierbij een belangrijk gemis is. Met de combinatie van ondergrens en bovengrens kunnen namelijk diverse planologische scenario's getoetst worden op hun kostenconsequenties. Zo kunnen bijvoorbeeld de kostenconsequenties van het beleidsvoornemen getoetst worden, om alleen kleine zandwinputten te creëren (door de onder- en bovengrens laag te kiezen) of om

---

* In het kostenmodel wordt als ondergrens de minimale zandproduktie gehanteerd. Middels de formules (3.2) en (3.3) uit paragraaf 3.3 wordt de relatie gelegd tussen de zandproduktie en de grootte van de zandwinput. Een dergelijke relatie kan ook gelegd worden voor een bovengrens voor de grootte van de zandwinput.

alleen zeer grote zandwinputten te creëren (door de onder- en bovengrens hoog te kiezen).

Model operationeel

- Dat het kostenmodel vraaggestuurd is, bepaalt in grote mate het karakter van het model. Het model geeft aan welke potentiële zandwinlocaties in produktie genomen moeten worden bij een gegeven vraag. Wanneer het kostenmodel aanbodgestuurd zou zijn zou er bij een gegeven vraag alleen nog bepaald moeten worden welke transportwijzen gekozen worden. Het model zou hierdoor aanzienlijk eenvoudiger van opzet zijn en het is zelfs denkbaar dat een handberekening zou volstaan.
- De invloed van het hanteren van een ondergrens voor de grootte van de zandproduktie en van het introduceren van een bovengrens op het beschikbaar komen van een oplossingsalgorithme wordt besproken in paragraaf 8.7.1.

De conclusie is dat, omdat een vraaggestuurd model veel beter overeenkomt met de werkelijkheid dan een aanbodgestuurd model, betreffend uitgangspunt gehandhaafd dient te blijven. Dit ondanks het feit, dat voor een aanbodgestuurd model een veel eenvoudiger oplossingsalgorithme of zelfs een handberekening volstaat. Het hanteren van een ondergrens voor de grootte van de zandwinput dient eveneens gehandhaafd te blijven en het in het kostenmodel opnemen van een bovengrens voor de grootte van de zandwinput verdient aanbeveling. Hoewel het introduceren van een bovengrens zekere gevolgen heeft voor de spline functie formulering en dus voor het beschikbaar komen van een oplossingsalgorithme, zal de bruikbaarheid van het model als beleidsondersteunend instrument dusdanig toenemen dat de introductie van een bovengrens gerechtvaardigd lijkt.

8.4 Er wordt geen specifieke korreldiameter gevraagd

Model waarheidsgetrouw

- Dat bij de vraag naar zand geen specifieke korreldiameter gevraagd wordt, is overeenkomstig de werkelijkheid. De potentiële zandwinlocaties zijn zodanig geselecteerd, dat ze over zand beschikken dat geschikt is voor ophoogdoeleinden. De gemiddelde korreldiameter van het zand is dan groter dan ca. 60 μ. Er is dus geen noodzaak om aanvullende eisen aan de gemiddelde korreldiameter van het zand te stellen.

Alternatief beleid mogelijk

- De korreldiameter van het gewonnen zand is niet van belang bij het afwegen van alternatief beleid.

Model operationeel

- Indien er desondanks een specifieke vraag is naar een bepaalde korreldiameter of naar een bepaalde spreiding in de korreldiameters, dan valt het probleem uiteen in een aantal deel-

problemen op basis van de gevraagde en aangeboden korreldiameters. De structuur van deze problemen zal dan gelijk zijn aan die van het oorspronkelijke probleem, alleen zullen de problemen kleiner van omvang zijn. Hierdoor zal de kans op het beschikbaar komen van een oplossingsalgorithme toenemen.

De conclusie is dat betreffend uitgangspunt gehandhaafd dient te blijven. Indien om bijzondere redenen een specifieke vraag naar een bepaalde korreldiameter optreedt, vergroot dit de kans op het beschikbaar komen van een oplossingsalgorithme.

8.5 Het zand wordt gewonnen middels een winzuiger

Model waarheidsgetrouw

- Zoals uit de tabellen 3 en 4 in paragraaf 2.3 blijkt, wordt het grootste deel van het zand gewonnen uit binnenwateren en groeven. De profielzuiger en de cutterzuiger zijn hierbij de meest geëigende winmiddelen. Betreffend uitgangspunt is dus overeenkomstig de werkelijkheid.
- Voor winning vanuit binnenwateren zal een aantal modelparameters moeten worden aangepast. Zo is er geen sprake meer van aankoopkosten van terreinen maar van een eventuele pachtsom of een heffing per m^3 gewonnen zand. Daarnaast zijn er geen kosten voor het graven van de put die nodig is om een drijvende winzuiger in te kunnen zetten. Verder bestaat de mogelijkheid om zwaarder materieel in te zetten*. De aanpassing van de modelparameters kan plaats vinden zonder dat de modelstructuur wijzigt.
- Anders ligt het bij ophoogzandwinning op zee. Hierbij is er sprake van een ander winmiddel (de sleephopperzuiger), het ontbreken van specifieke winlocaties, geen aankoopkosten van terreinen, deels van varend transport (mogelijk naar verschillende walbestemmingen) en van kosten voor ontzilting. Door deze andere kostenopbouw zal de modelstructuur wijzigen en het huidige kostenmodel is dan ook niet geschikt als beleidsondersteunend instrument voor ophoogzandwinning op zee.

Alternatief beleid mogelijk

- Door het uitgangspunt, dat het ophoogzand middels een winzuiger gewonnen wordt, bestaat de mogelijkheid om winning vanuit landlocaties en vanuit binnenwateren af te wegen.
- Het alternatief van ophoogzandwinning op zee is hierbij uitgesloten, hetgeen geen bezwaar is omdat het kostenmodel voor afweging van deze wijze van winning toch niet geschikt is.

---

* Bij de bepaling van de kostenparameters is uitgegaan van een maximale leidingdiameter O 0,60 m, hetgeen de grootste over land nog redelijk hanteerbare installatie is. Bij aan- en afvoer over water kunnen grotere leidingdiameters (bijvoorbeeld O 0,90 m) overwogen worden.

Model operationeel

- De variabele kosten van de winning met behulp van een winzuiger zijn opgenomen in de transportkosten. In het kostenmodel zijn als kosten van het inzetten van een winzuiger alleen vaste kosten opgenomen. Deze vaste kosten zijn bovendien onafhankelijk verondersteld van de korreldiameter van het te winnen zand. De invloed van het inzetten van een winzuiger op het beschikbaar komen van een oplossingsalgorithme is daarom gering.

De conclusie is dat, omdat in het kostenmodel alleen de winning vanuit landlocaties en eventueel vanuit binnenwateren beschouwd wordt, betreffend uitgangspunt gehandhaafd dient te blijven. De gevolgen voor het beschikbaar komen van een oplossingsalgorithme zijn hierbij gering.

8.6 Het zand wordt getransporteerd per vrachtauto of per pijpleiding

Model waarheidsgetrouw

- Voor zandwinning vanuit landlocaties of vanuit binnenwateren zijn de vrachtauto en de pijpleiding de geëigende transportmiddelen. Transport per spoor komt alleen voor bij ophoogzand voor eigen gebruik van de Nederlandse Spoorwegen en transport per binnenvaartschip is zeldzaam. Dat zowel het transport per spoor als per binnenvaartschip weinig toegepast worden heeft de volgende reden:
  . Het transport van ophoogzand vindt over het algemeen plaats over beperkte afstand; aan de vraag naar ophoogzand wordt veelal voorzien vanuit zandwinlocaties in de regio. Hierdoor vallen de kostenvoordelen die behaald kunnen worden door transport per spoor of per schip weg tegen de extra overslagkosten. Zelfs wanneer het vraagpunt zich in de directe nabijheid van een spoor- of vaarweg bevindt moet het zand nog worden overgeslagen. Het kan immers niet direct vanuit de spoorwagon of het schip in het werk worden gebracht.

Alternatief beleid mogelijk

- De mogelijkheid wordt geboden om de invloed van de twee transportwijzen op de totale kosten van de ophoogzandvoorziening af te wegen. Zo kan bijvoorbeeld de invloed van het beleidsvoornemen om transport per vrachtwagen vanwege de verkeersveiligheid te weren (door de kosten zeer hoog te stellen) worden bepaald.
- De alternatieve transportwijzen (transport per spoor en per schip) kunnen niet worden afgewogen. Omdat deze transportwijzen voor ophoogzand vrij zeldzaam zijn, is dit geen groot bezwaar. Indien noodzakelijk bestaat de mogelijkheid betreffende transportwijzen in het kostenmodel op te nemen zonder dat de structuur van het model wijzigt.

Model operationeel

- Het opnemen van het transport per pijpleiding in het kostenmodel is de voornaamste oorzaak voor de gecompliceerdheid en de omvang van het model en dus voor de noodzaak voor het toepassen van een specifiek voor het probleem ontwikkeld oplossingsalgorithme. Daarom heeft dit uitgangspunt grote invloed op het beschikbaar komen van een oplossingsalgorithme (zie paragraaf 8.7.3).
- Indien transport per spoor en per schip in het model worden opgenomen zal het aantal variabelen en beperkingen toenemen, hetgeen het beschikbaar komen van een oplossingsalgorithme negatief beïnvloedt. De structuur van het model zal echter niet wijzigen.

De conclusie is dat betreffend uitgangspunt gehandhaafd dient te blijven omdat voor het ophoogzandtransport vanuit landlocaties en binnenwateren de vrachtauto en de pijpleiding de aangewezen transportmiddelen zijn. Dat met name het opnemen van pijpleidingtransport in het kostenmodel het beschikbaar komen van een oplossingsalgorithme bemoeilijkt, moet hierbij geaccepteerd worden. Een kostenmodel waarin het transport per pijpleiding niet is opgenomen, heeft als beleidsondersteunend instrument een veel geringere betekenis. Indien transport per pijpleiding het beschikbaar komen van een oplossingsalgorithme blokkeert, kan overwogen worden het transport per pijpleiding op een vereenvoudigde wijze in het kostenmodel op te nemen. Indien de specifieke situatie dit vereist kunnen transport per spoor of per schip in het kostenmodel worden opgenomen zonder dat de structuur van het model wijzigt. Het aantal variabelen en beperkingen zal echter toenemen, hetgeen een negatieve invloed heeft op het beschikbaar komen van een oplossingsalgorithme.

8.7 Kostencategorieën

8.7.1 De aankoopkosten van de benodigde terreinen

Model waarheidsgetrouw

- De aankoopkosten van het benodigde terrein zijn in veel gevallen een onderdeel van de totale kosten van de ophoogzandvoorziening. Zoals uit paragraaf 2.5 blijkt, zijn er echter situaties waarin het terrein niet wordt aangekocht. In die gevallen dient een bepaald bedrag betaald te worden (dat niet evenredig behoeft te zijn met de oppervlakte van het benodigde terrein) of dient een andere tegenprestatie geleverd te worden (zoals de inrichting van de achterblijvende waterplas of zelfs van een ander terrein).
- Voor zandwinning vanuit binnenwateren is er geen sprake van aankoopkosten; er zal dan sprake zijn van een bepaalde pachtsom of van een heffing per m^3 gewonnen zand.
- De aankoop- of huurkosten van het terrein ten behoeve van een depot (bij transport per vrachtwagen) zijn niet in het model opgenomen. De grootte van het depot is afhankelijk van de

putproduktie en van de transportcapaciteit, een parameter die niet in het kostenmodel is opgenomen.
- De aankoop- of huurkosten van het terrein ten behoeve van de plaatsing van het pompstation en het deel van de persleiding dat zich op de winlocatie bevindt (bij transport per pijpleiding) zijn niet in het model opgenomen.
- Er wordt geen maximum gesteld aan het oppervlak van de winput; zie hiervoor paragraaf 8.3.
- De winput wordt geschematiseerd als een omgekeerde, eventueel afgeknotte, piramide met een vierkant grondvlak terwijl er vele vormen mogelijk zijn (fig. 31).

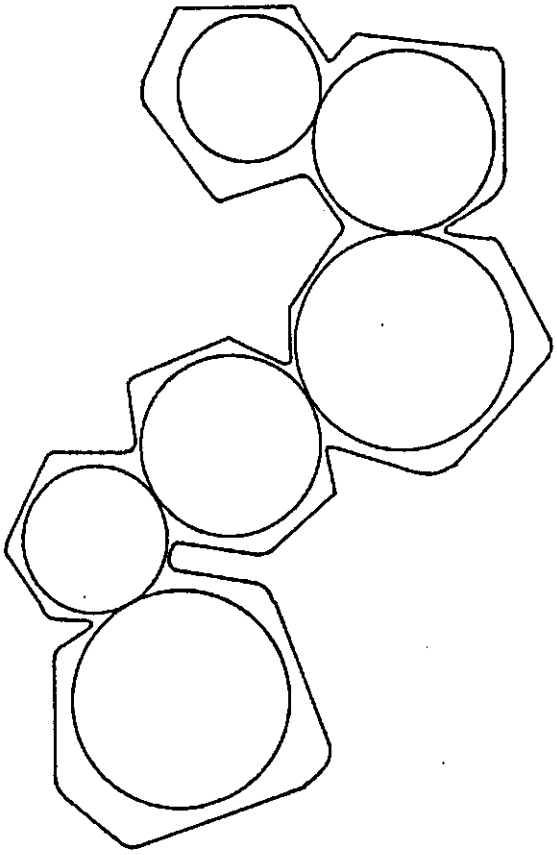


Fig. 31 Mogelijke inrichting van na zandwinning achterblijvende waterplassen ([3])

- In het kostenmodel wordt geen maximum gesteld aan de diepte van de achterblijvende waterplas. De diepte wordt bepaald door de dikte van de niet uit zand bestaande bovenlaag, de dikte van de zandlaag en de grondwaterstand, een parameter die niet in het model is opgenomen. In de praktijk zal er veelal wel een maximale waterdiepte geëist worden uit biologische en recreatieve overwegingen (spronglaageffect, ondiep zwemwater).

- De kosten van het verwijderen van de niet uit zand bestaande bovenlaag worden in het kostenmodel in de grondprijs opgenomen, terwijl ze niet rechtevenredig zijn met de oppervlakte van de winput.
- De afwerkingskosten van de achterblijvende waterplas en van de oevers zijn niet in het model opgenomen. Ze zijn met name afhankelijk van de oeverlengte, die een functie is van de putproduktie.
- Er is in het model sprake van slechts één talud terwijl in werkelijkheid verspringende taluds geëist kunnen worden, bijvoorbeeld voor zwemwater (fig. 32).

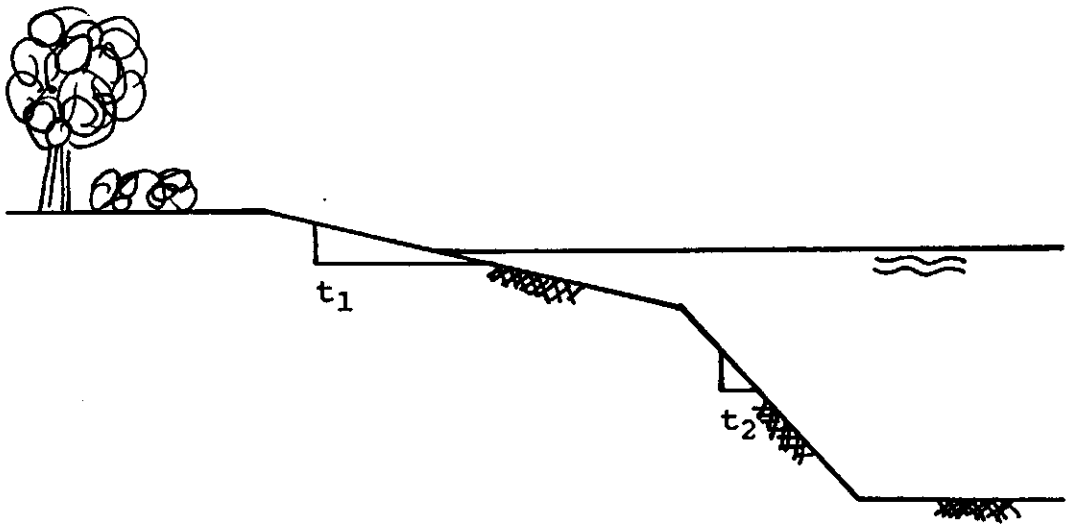


Fig. 32 Verschillende taluds bij gebruik van de na zandwinning achterblijvende waterplas als zwemwater

- Bij de spline functie benadering is de invloed van het aantal spline intervallen, van het gekozen afwijkingscriterium (relatieve of absolute afwijking) en van de grootte daarvan vooralsnog onbekend.

Alternatief beleid mogelijk

- Het opnemen van de aankoopkosten van de terreinen in het kostenmodel is van groot belang voor de betekenis van het kostenmodel als beleidsondersteunend instrument. Door de grondprijs van de diverse potentiële zandwinlocaties te variëren kan de invloed van een beleidsmatige keuze voor bepaalde locaties op de totale kosten van de ophoogzandvoorziening worden aangetoond. Zo kan de invloed van het uitsluiten van een bepaalde locatie (bijvoorbeeld uit milieuoverwegingen) worden vastgesteld. De kosten bij beschikbaarheid van betreffende zandwinlocatie kunnen worden vergeleken met de kosten bij uitsluiten van betreffende locatie (door de grond-

prijs zo hoog te kiezen dat betreffende locatie zeker niet in produktie genomen wordt).

Model operationeel

- De volgende modelparameters hebben betrekking op de aankoopkosten van de terreinen:

  a(j) : vraag naar zand van vraagpunt j [m^3]
  d(i) : dikte van de laag winbaar zand [m]
  K : aantal spline functie intervallen [-]
  m : totaal aantal vraagpunten [-]
  n : totaal aantal zandwinputten [-]
  pg(i) : grondprijs [f/m^2]
  rq(i,0): minimaal te winnen hoeveelheid zand in geval de winput in produktie genomen wordt [m^3]

  ρ : relatieve afwijking $\frac{RS(q)-R(q)}{R(q)} < \rho$ [-]
  of
  δ : absolute afwijking RS(q)-R(q)
  t(i) : talud van de winput [-]
  v(i) : dikte van de niet uit zand bestaande bovenlaag [m]

  Voor het proefgebied betreft het 150 van de in totaal 78023 modelparameters waarvan er 49 ook betrekking hebben op andere kostencategorieën.
- De volgende modelgrootheden zijn hiervan afgeleid:

  gam(i,k): R'(rq(i,k))-R'(rq(i,k-1))
  kq(i,k) : zandproduktie waarbij het k-de knikpunt in de spline functie optreedt [m^3]
  Q(i,k) : QA-kq(i,k) [m^3]
  QA : totale vraag naar zand [m^3]
  R(i,k) : lineaire benadering van de functie R(i,q) [m^2]
  R'(i,k) : $\frac{dR(i,q)}{dq}$ [m^{-1}]
  rq(i,1) : hoeveelheid te winnen zand waarbij de winput overgaat van omgekeerde piramidevorm in afgeknotte omgekeerde piramidevorm [m^3]
  rq(i,k) : hoeveelheid zand waarbij de benadering RS(rq(i,k)) raakt aan de functie R(rq(i,k)) in het k-de raakpunt [m^3]

  Voor het proefgebied betreft het 571 van de in totaal 7259 afgeleide grootheden waarvan er 1 ook betrekking heeft op andere kostencategorieën.
- De volgende modelvariabelen hebben betrekking op de aankoopkosten van de terreinen:

  d1(i) : dummievariabele die aangeeft of winput i in produktie genomen wordt [-]
  d4(i,k): dummievariabele die aangeeft of de putproduktie van winput i groter is dan kq(i,k) [-]
  qw(i,j) : hoeveelheid zand die per vrachtwagen getransporteerd wordt van winput i naar vraagpunt j [m^3]

qb(i,j) : hoeveelheid zand die per pijpleiding getransporteerd wordt van winput i naar vraagpunt j [m^3]
qx(i,k): hoeveelheid zand die méér geproduceerd wordt dan de hoeveelheid kq(i,k) [m^3]

Voor het proefgebied betreft het 1995 van de in totaal 3648 modelvariabelen waarvan er 1805 ook betrekking hebben op andere kostencategorieën.

- De parameters, afgeleide grootheden en variabelen die betrekking hebben op de aankoopkosten van de terreinen maken deel uit van alle 1928 beperkingen waarvan er 1719 ook betrekking hebben op andere kostencategorieën.
- Een aantal van de ontbrekende aspecten kan in het kostenmodel worden opgenomen zonder dat dit grote invloed heeft op het beschikbaar komen van een oplossingsalgorithme. Voor andere aspecten zoals een bovengrens voor de grootte van de winput, de vorm van de winput, de maximale waterdiepte, de kosten van het verwijderen van de niet uit zand bestaande bovenlaag en de verspringende taluds geldt dat deze aspecten een zekere invloed hebben op de spline functie formulering en dus ook op het beschikbaar komen van een oplossingsalgorithme. Hoe groot deze invloed is moet van aspect tot aspect bezien worden.

De conclusie is dat, omdat de aankoopkosten van de terreinen een wezenlijke rol spelen bij het afwegen van beleidsalternatieven, ze in het model moeten blijven opgenomen. Dat het opnemen van de aankoopkosten van de terreinen een zekere invloed heeft op het beschikbaar komen van een oplossingsalgoritme moet hierbij geaccepteerd worden. De aankoopkosten van de terreinen zijn in vereenvoudigde vorm in het kostenmodel opgenomen. Het verdient aanbeveling te onderzoeken wat de invloed van de diverse vereenvoudigingen is. Indien ze aanzienlijk is, zal het kostenmodel moeten worden uitgebreid hetgeen de kans op het beschikbaar komen van een oplossingsalgorithme negatief beïnvloedt.

8.7.2 De aanlegkosten

Model waarheidsgetrouw

- Zoals in paragraaf 3.4 is aangegeven, zijn de aanlegkosten van de zandwinput alsvolgt samengesteld:
  . De kosten van het graven van de put, nodig om de drijvende winzuiger te kunnen inzetten.
  . De kosten van aanleg en opruimen van de winzuiger.
  . De kosten van aanleg en opruimen van de drijvende pijpleiding.
  . De kosten van aanleg en opruimen van het deel van de pijpleiding dat zich op de winlocatie bevindt.

  Voor zandwinning vanuit landlocaties komen deze kosten overeen met de werkelijk optredende kosten. Voor winning vanuit binnenwateren is er sprake van een andere kostenopbouw (zie paragraaf 8.5).

Alternatief beleid mogelijk

- De aanlegkosten zijn met name van belang bij het afwegen van winning vanuit landlocaties en vanuit binnenwateren. Zo kunnen de kostenconsequenties worden vastgesteld van het beleidsvoornemen om het zand alleen vanuit binnenwateren te winnen. De totale kosten van de ophoogzandwinning bij beschikbaarheid van de landlocaties kunnen vergeleken worden met de totale kosten bij uitsluiten van betreffende locaties (een scenario dat overigens ook gerealiseerd kan worden door de grondprijs voor de landlocaties zeer hoog te stellen)

Model operationeel

- De volgende modelparameters hebben betrekking op de aanlegkosten van de zandwinputten:

  a(j) :vraag naar zand van vraagpunt j [m^3]
  CA(i): vaste aanlegkosten van de zandwinput i [f]
  CB(i): vaste winningskosten bij transport per pijpleiding vanuit winput i [f]
  m : totaal aantal vraagpunten [-]
  n : totaal aantal zandwinputten [-]

  Voor het proefgebied betreft het 87 van de in totaal 78023 modelparameters, waarvan er 68 ook betrekking hebben op andere kostencategorieën.
- De volgende modelgrootheid is hiervan afgeleid:

  QA: totale vraag naar zand [m^3]

  Voor het proefgebied betreft het 1 van de in totaal 7259 afgeleide grootheden, die bovendien ook betrekking heeft op andere kostencategorieën.
- De volgende modelvariabelen hebben betrekking op de aankoopkosten van de terreinen:

  d1(i) : dummievariabele die aangeeft of winput i in produktie genomen wordt [-]
  qw(i,j): hoeveelheid zand die per vrachtwagen getransporteerd wordt van winput i naar vraagpunt j [m^3]
  qb(i,j): hoeveelheid zand die per pijpleiding getransporteerd wordt van winput i naar vraagpunt j [m^3]

  Voor het proefgebied betreft het 1805 van de totaal 3648 modelvariabelen die allen ook betrekking hebben op andere kostencategorieën.
- De parameters, afgeleide grootheden en variabelen die betrekking hebben op de aanlegkosten van de zandwinputten maken deel uit van 1833 van de in totaal 1928 beperkingen die allen ook betrekking hebben op andere kostencategorieën.

De conclusie is dat de aanlegkosten in het kostenmodel moeten blijven opgenomen. Voor winning vanuit landlocaties komen ze overeen met de werkelijkheid, voor winning vanuit binnenwateren zullen de kostenparameters moeten worden aangepast. De structuur van het kostenmodel wijzigt hierdoor niet en er is geen invloed op het beschikbaar komen van een oplossingsalgorithme.

8.7.3 De kosten van winning en transport bij transport per vrachtwagen

Model waarheidsgetrouw

- Zoals in paragraaf 3.5 is aangegeven worden hieronder de volgende kosten verstaan:
  . Het verschil tussen de vaste winningskosten bij transport per vrachtwagen en de vaste winningskosten bij transport per pijpleiding.
  . De variabele winningskosten; dit zijn de kosten van de huur van het winmaterieel, de brandstofkosten en de personeelskosten.
  . De variabele transportkosten; dit zijn de kosten van de huur van het transportmaterieel, de brandstofkosten en de personeelskosten.

  De invloed van de korreldiameter op de variabele winningskosten wordt in het model verwaarloosd. Deze aanname lijkt gerechtvaardigd en de kosten komen redelijk overeen met de werkelijk optredende kosten.
- In het kostenmodel zijn de kosten van schade aan het wegennet niet opgenomen. Deze kosten kunnen als variabele transportkosten in het model worden opgenomen zonder dat de structuur van het model wijzigt.

Alternatief beleid mogelijk

- De mogelijkheid bestaat om de invloed van de brandstof- en personeelskosten op de keuze voor een bepaalde transportwijze (vrachtwagen of pijpleiding) en op de totale kosten van de ophoogzandwinning vast te stellen.
- Het is mogelijk om de kostenconsequenties van het ontmoedigen van het transport per vrachtwagen omwille van de verkeersveiligheid vast te stellen. Hiertoe kan voor het transport per vrachtwagen een speciale heffing worden geïntroduceerd.

Model operationeel

- De volgende modelparameters hebben betrekking op de kosten van winning en transport bij transport per vrachtwagen:

  a(j) : vraag naar zand van vraagpunt j [m^3]
  CB(i) : vaste winningskosten bij transport per pijpleiding vanuit winput i [f]
  CW(i) : vaste winningskosten bij transport per vrachtwagen vanuit winput i [f]
  dw(i,j): gewogen transportafstand van winput i naar vraagpunt j [km]
  m : totaal aantal vraagpunten [-]
  n : totaal aantal zandwinputten [-]

  Voor het proefgebied betreft het 980 van de in totaal 78023 modelparameters waarvan er 70 ook betrekking hebben op andere kostencategorieën.
- De volgende modelgrootheden zijn hiervan afgeleid:

pw(i,j): prijs van het transport per vrachtwagen van winput i naar vraagpunt j [f/m^3]
QA : totale vraag naar zand [m^3]

Voor het proefgebied betreft het hier 894 van de in totaal 7259 afgeleide grootheden waarvan er 1 ook betrekking heeft op andere kostencategorieën.

- De volgende modelvariabelen hebben betrekking op de kosten van winning en transport bij transport per vrachtwagen:

  d2(i) : dummievariabele die aangeeft of er vanuit winput i transport per vrachtwagen plaatsvindt [-]
  qw(i,j): hoeveelheid zand die per vrachtwagen wordt getransporteerd van winput i naar vraagpunt j [m^3]

  Voor het proefgebied betreft het 912 van de in totaal 3648 modelvariabelen waarvan er 893 ook betrekking hebben op andere kostencategorieën
- De parameters, afgeleide grootheden en variabelen die betrekking hebben op de kosten van winning en transport bij transport per vrachtwagen maken deel uit van 199 van de in totaal 1928 beperkingen waarvan er 180 ook betrekking hebben op andere kostencategorieën.

De conclusie is dat de kosten van winning en transport bij transport per vrachtwagen, zoals ze in het kostenmodel zijn opgenomen, redelijk overeen komen met de werkelijk optredende kosten. Indien de verwachting is dat de kosten van schade aan het wegennet substantieel zijn, zullen ze in het model moeten worden opgenomen. Verder is in deze paragraaf aangetoond dat de invloed van de kosten van winning en transport bij transport per vrachtwagen op het beschikbaar komen van een oplossingsalgorithme gering is.

8.7.4 De kosten van winning en transport bij transport per pijpleiding

Model waarheidsgetrouw

- Zoals in paragraaf 3.6 is aangegeven worden hieronder de volgende kosten verstaan:
  . De vaste transportkosten; dit zijn de kosten van aanleg en afvoeren van het leidingennet inclusief de tussenstations.
  . De variabele winningskosten; dit zijn de kosten van huur van het materieel, de brandstofkosten en de personeelskosten.
  . De variabele transportkosten; dit zijn de kosten van huur van het materieel, de brandstofkosten en de personeelskosten.

  De invloed van de korreldiameter op de variabele winningskosten wordt in het model verwaarloosd. Deze aanname lijkt gerechtvaardigd en de kosten komen redelijk overeen met de werkelijk optredende kosten.
- Indien het water waarin het zand getransporteerd wordt problemen voor de lokale waterhuishouding op het stort met zich meebrengt, kan geëist worden dat het water teruggepompt wordt naar de zandwinput of naar een andere locatie verpompt wordt.

De kosten hiervan zijn niet in het kostenmodel opgenomen. Ze kunnen echter als vaste en variabele winningskosten in het model worden opgenomen zonder dat de structuur wijzigt.

Alternatief beleid mogelijk

- De mogelijkheid bestaat om de invloed van de brandstof- en personeelskosten op de keuze voor een bepaalde transportwijze (vrachtwagen of pijpleiding) en op de totale kosten van de ophoogzandwinning vast te stellen.

Model operationeel

- De volgende modelparameters hebben betrekking op de kosten van winning en transport bij transport per pijpleiding:

  a(j) : vraag naar zand van vraagpunt j [m^3]
  α(i) : kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand van winput i [f/km]
  coef(i,s,j): 0-1 coëfficiënt die aangeeft of traject s deel uitmaakt van het leidingtracé dat winput i met vraagpunt j verbindt [-]
  D(s) : lengte van leidingtraject s [km]
  δ(i) : kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand van winput i [f/km]
  m : totaal aantal vraagpunten [-]
  n : totaal aantal zandwinputten [-]
  S : totaal aantal pijpleidingtrajecten [-]

  Voor het proefgebied betreft het 76972 van de in totaal 78023 modelparameters waarvan er 49 ook betrekking hebben op andere kostencategorieën.
- De volgende modelgrootheden zijn hiervan afgeleid:

  CT(i,s): vaste kosten van traject s van het leidingennet van winput i [f]
  DB(i,j): $\sum_{s=1}^{S} (coef(i,s,j)D(s))$, lengte van het leidingtracé dat winput i met vraagpunt j verbindt [km]
  pb(i,j): prijs van het winnen van het zand vanuit winput i en het transporteren naar vraagpunt j [f/m^3]
  QA : totale vraag naar zand [m^3]

  QB(i,s): $\sum_{j=1}^{m} (coef(i,s,j)a(j))$

  Voor het proefgebied betreft het 5055 van de in totaal 7259 afgeleide grootheden waarvan er 1 ook betrekking heeft op andere kostencategorieën.
- De volgende modelvariabelen hebben betrekking op de kosten van winning en transport bij transport per pijpleiding:

d3(i,s): dummievariabele die aangeeft of traject s van het leidingennet van winput i gebruikt (en dus aangelegd) wordt [-]
qb(i,j): hoeveelheid zand die per pijpleiding getransporteerd wordt van winput i naar vraagpunt j [m^3]

Voor het proefgebied betreft het 2527 van de in totaal 3648 modelvariabelen waarvan er 893 ook betrekking hebben op andere kostencategorieën.

De conclusie is dat de kosten van winning en transport bij transport per pijpleiding, zoals ze in het kostenmodel zijn opgenomen, redelijk overeenkomen met de werkelijk optredende kosten. Indien de specifieke situatie de aanleg van retourleidingen vereist zullen de kosten hiervan in het model moeten worden opgenomen omdat ze substantieel zullen zijn. Verder is in deze paragraaf aangetoond dat de kosten van winning en transport bij transport per pijpleiding verreweg de grootste invloed hebben op het beschikbaar komen van een oplossingsalgorithme.

8.8 Organisatorische aspecten en kosten/baten

8.8.1 Organisatorische aspecten

Het onderzochte kostenmodel heeft tot functie het beleid t.a.v. de verlening van ontgrondingsvergunningen te ondersteunen. Zoals uit paragraaf 2.5 blijkt, is de provinciale overheid voor alle locaties m.u.v. de zee en de rijkswateren de vergunningverlenende instantie. In deze paragraaf wordt daarom ingegaan op de organisatorische aspecten van het gebruik van het kostenmodel binnen de provinciale overheidsorganisaties.

Op het eerste gezicht lijkt het kostenmodel ook een functie te hebben als bedrijfskundig instrument voor de particuliere ondernemer. Het is voor deze toepassing echter niet geschikt:

- De particuliere ondernemer moet een specifieke vraag naar ophoogzand dekken voor een beperkt aantal projecten; hij is als ophoogzandleverancier niet bij alle projecten in een bepaald gebied betrokken.
- De projecten waarvoor de particuliere ondernemer het ophoogzand moet leveren kunnen zich bovendien in andere regio's bevinden, waardoor betreffende ondernemer geen geïntegreerd winningsbeleid kan voeren.
- Aan de particuliere ondernemer is veelal slechts voor een beperkte periode bekend hoeveel ophoogzand hij moet leveren, namelijk de periode waarover zijn leveringen en reeds afgesloten contracten lopen. Hij is er veelal niet van op de hoogte voor welke toekomstige projecten hij ophoogzand zal moeten leveren en kan dus geen geïntegreerd winningsbeleid voeren op basis van deze wetenschap.
- De particuliere ondernemer is geen beleidsmaker; hij zal uit de door de overheid geboden mogelijkheden t.a.v. de zandwinlocaties, de grootte van de terreinen en de transportwijzen de meest economische keuze maken. Door het beperkte aantal

mogelijkheden zal deze keuze veelal op basis van een handberekening gemaakt kunnen worden.

Zinvol gebruik van het kostenmodel door de provinciale overheden vereist invulling van de volgende aspecten:

Beleidsalternatieven

Met het kostenmodel kunnen de kostenconsequenties van bepaalde beleidsalternatieven worden aangetoond; het model is niet geschikt om er de beleidsalternatieven zelf mee te formuleren.

Hoewel de provincies beschikken over ontgrondingsplannen c.q. nota's en/of structuurplannen is het huidige vergunningenbeleid nog in hoofdzaak een ad hoc beleid. De geformuleerde richtlijnen hebben over het algemeen een technisch karakter (zie paragraaf 2.6). De provinciale overheden zullen dus een aantal beleidsalternatieven moeten formuleren. Deze kunnen betrekking hebben op:

- Het definiëren van de verschillende deelgebieden
- Het aanwijzen van de potentiële zandwinlocaties binnen de deelgebieden
- De bestemming van de winlocaties na winning
- Centrale of decentrale winning binnen de deelgebieden
- Minimale en maximale grootte van de winputten
- Voorkeur voor bepaalde winmethoden en transportwijzen
- Beïnvloeden van de vraag door:
  . bouwvergunningen aan bepaalde projecten te onthouden
  . invloed op de locatiekeuze van de geplande projecten uit te oefenen
  . het stimuleren of afremmen van partiële ophoging
  . het stimuleren of afremmen van het gebruik van alternatieve ophogingsmaterialen

Gegevens

Zowel voor het formuleren van de beleidsalternatieven als voor het gebruik van het kostenmodel is een groot aantal gegevens vereist:

T.a.v. de potentiële winlocaties:

- Gegevens t.a.v. het huidige gebruik van de terreinen
- Toekomstig gebruik (bestemmingsplannen)
- Topografische gegevens
- Gedetailleerde grondgegevens
- Eigendomsverhoudingen
- Grondprijzen en eventuele onteigeningskosten

T.a.v. de winning en het transport:

- De bereikbaarheid van de potentiële zandwinlocaties en vraagpunten
- De mogelijke winmethoden
- Wegennet en afstanden

- Eigendomsverhoudingen (i.v.m. de eventuele aanleg van pijpleidingnetten)
- Huurprijzen van het materieel
- Personeelskosten
- Brandstofprijzen

T.a.v. de vraag:

- Locaties van de vraagpunten
- Gevraagde hoeveelheden per vraagpunt
- Periode waarin de vraag per vraagpunt optreedt
- Nauwkeurigheid van de voorspelde vraag

De provincies kunnen momenteel reeds over een aantal van deze gegevens beschikken. Voor andere gegevens moet nader onderzoek gedaan worden en moet overleg plaatsvinden met de rijksoverheid, met de locale overheden en met het particuliere bedrijfsleven.

Computer

Het minimaliseringsprobleem, waaruit het kostenmodel bestaat, is voor een onderzoeksgebied van enige omvang dermate groot dat het alleen met een computer kan worden opgelost. Bovendien zal dit gezien het te verwachten geheugenbeslag en de te verwachten rekentijd een mainframe computer moeten zijn. De provinciale overheden hebben doorgaans toegang tot dergelijke computers. Wanneer het probleem dermate groot is dat bij gebruik van een mainframe computer onacceptabel lange rekentijden optreden, kan gebruik worden gemaakt van zgn. supercomputers die bij een aantal gespecialiseerde rekencentra staan opgesteld en waar rekentijd gehuurd kan worden.

Programmatuur

Onderhavig onderzoek heeft helaas geen direct toepasbare programmatuur opgeleverd. Omdat met de programmatuur gebaseerd op het ICW-algorithme wel de oplossing voor een deelprobleem is verkregen, lijkt het gerechtvaardigd deze programmatuur verder te ontwikkelen. Dat er standaardprogrammatuur op de markt is, waarmee problemen van een omvang als die van het volledige proefgebied kunnen worden opgelost, moet worden betwijfeld. Men zal gedwongen zijn tot het ontwikkelen van programmatuur voor het specifieke probleem. Het verdient echter aanbeveling om te onderzoeken welke programmatuur er reeds ontwikkeld is voor gelijkvormige gemengd geheeltallige lineaire transportproblemen.

Personele organisatie

Het formuleren van beleidsalternatieven behoort tot de taken van de provinciale overheid; hiervoor zal normaal gesproken dan ook geen aanpassing van de bestaande organisatiestructuur nodig zijn.

Het verzamelen van de gegevens zal binnen de provinciale overheid gecoördineerd moeten worden. Indien de bestaande organisatiestructuren en personeelsbezettingen niet volstaan, dan zullen ze gecreëerd moeten worden.

Het verder ontwikkelen van de programmatuur lijkt buiten de directe taakstelling van de provinciale overheid te vallen. Er zijn echter voldoende onderzoeksinstituten die een dergelijke taak op zich zouden kunnen nemen.

Het gebruik van kostenmodel en bijbehorende programmatuur vereist een zekere onderzoekservaring. Indien deze binnen de provinciale overheden niet voorhanden is, kan ze aangetrokken worden. Hierbij moet dan wel in een goede communicatie voorzien zijn met de instantie die de programmatuur ontwikkeld heeft. Een andere mogelijkheid is het gebruik in handen van een externe operator te geven. Het instituut dat de programmatuur ontwikkeld heeft, zal hierbij de voorkeur genieten. Er moet immers communicatie plaatsvinden tussen de provincie als beleidsinstantie en leverancier van gegevens en de operator. Indien de operator een andere instantie is dan de ontwikkelaar van de programmatuur, zijn er drie instanties bij betrokken, hetgeen de communicatie bemoeilijkt.

De conclusie is dat het voor de provinciale overheidsorganisaties zeer wel mogelijk is om het kostenmodel te gebruiken, mits de voorgaande aspecten ingevuld worden. Een voorbehoud moet hierbij gemaakt worden voor het beschikbaar komen van een werkend oplossingsalgorithme met bijbehorende programmatuur.

8.8.2 Kosten/baten

Kosten

Aan ieder van voornoemde in te vullen aspecten zijn kosten verbonden. Een deel van deze kosten zal niet direct toewijsbaar zijn omdat ze opgaan in de kosten van een bestaande organisatie; dit zal met name het geval zijn met een deel van de personeelskosten. Andere kosten kunnen direct worden gekoppeld aan het gebruik van het kostenmodel; het betreft dan met name investeringskosten en kosten van prestaties die uitbesteed worden. De hoogte van deze kosten zal van provincie tot provincie verschillen. Bovendien zijn de kosten voor een deel afhankelijk van het aantal problemen en alternatieven dat beschouwd wordt. In tabel 6 wordt een schatting gegeven van de jaarlijkse kosten per provincie. Hierbij is uitgegaan van de volgende aannamen:

- De kosten voor ontwikkeling van de programmatuur zijn eenmalig en worden door de 12 provincies gedeeld.
- De te beschouwen periode is 5 jaar; de invoergegevens worden eenmalig voor deze periode verzameld en jaarlijks bijgesteld.
- De programmatuur wordt in deze periode afgeschreven.
- De kosten van het formuleren van beleidsalternatieven blijven buiten beschouwing; de provinciale overheid zou ook beleidsvoornemens moeten maken, zonder dat het kostenmodel in gebruik is.
- Er worden per provincie 5 gebieden beschouwd van een omvang die vergelijkbaar is met die van het volledige proefgebied (bijlage II).

- Per gebied worden jaarlijks 20 beleidsalternatieven doorgerekend.
- Renteverliezen en prijsstijgingen blijven buiten beschouwing

Eenmalige kosten:	
ontwikkeling programmatuur 1500 u à f 200,- (/12)	f 25.000,-
verzamelen gegevens 5 x 200 u à f 100,-	f 200.000,-
	+
totaal	f 225.000,-
Jaarlijkse kosten:	
eenmalige kosten f 225.000,- / 5	f 45.000,-
bijstellen gegevens 5 x 50 u à f 100,-	f 25.000,-
beleidsalternatieven 5 x 20 x 15 u à f 100,-	f 150.000,-
computerkosten 5 x 20 x f 400,-	f 40.000,-
totaal	f 270.000,-

Tabel 6 Kostenschatting van het gebruik van het kostenmodel per provincie per jaar

Bovenstaande schatting is bedoeld om van de kosten die voortvloeien uit het gebruik van het kostenmodel de orde van grootte aan te geven. Er wordt hierbij geen aanspraak gemaakt op een grote nauwkeurigheid.

Baten

Gebruik van het kostenmodel kan resulteren in baten die deels kwantificeerbaar en deels niet-kwantificeerbaar zijn.

Kwantificeerbare baten

- Het beperken van de grondkosten door een optimale keuze van de zandwinlocaties.
- Het beperken van de winkosten door een optimale keuze van de zandwinlocaties.
- Het beperken van de transportkosten door een optimale keuze van de zandwinlocaties en de transportwijzen.

Het inschatten van de kwantificeerbare baten is bijzonder moeilijk:

- De baten kunnen pas echt worden bepaald nadat met behulp van het kostenmodel de keuze voor de zandwinlocaties en de transportwijzen gemaakt is.
- De baten verschillen van provincie tot provincie en binnen de provincie van gebied tot gebied.
- De baten verschillen van periode tot periode.

- Baten voor één betrokkene kunnen samenvallen met kosten voor een andere betrokkene.
- Een aanzienlijk deel van de baten zal ten goede komen aan andere betrokkenen dan de provinciale overheden; door gebrek aan informatie over deze betrokkenen zijn deze baten nog moeilijker in te schatten.

Niet-kwantificeerbare baten

- Een beter planologisch beleid.
- Een beter milieubeleid
- Een beter verkeersbeleid (verkeersveiligheid en verkeers-overlast in relatie tot het transport).

De conclusie is dat de baten van het gebruik van het kostenmodel nauwelijks kunnen worden gekwantificeerd en zodoende ook niet kunnen worden afgewogen tegen de kosten (die ook moeilijk zijn vast te stellen). Gezien de ordegrootte van de jaarlijkse kosten en de mogelijke baten lijkt gebruik van het kostenmodel echter gerechtvaardigd.

9. Conclusies en aanbevelingen

Op basis van het afstudeeronderzoek zijn een aantal conclusies getrokken. De belangrijkste conclusies zijn hierbij natuurlijk de conclusies waarmee aan de doelstelling beantwoord wordt, de conclusies waarmee wordt vastgesteld in hoeverre het kostenmodel een bijdrage kan leveren aan de provinciale beleidsvorming t.a.v. de verlening van ontgrondingsvergunningen. Bij het weergeven van de conclusies is dezelfde volgorde gehanteerd als die van de onderdelen van het verslag, waaruit ze afkomstig zijn; ze worden ook nog geïntegreerd tot een slotconclusie.

Daarnaast wordt een aantal aanbevelingen gedaan voor verdere studie.

Conclusies

- De winning van ophoogzand legt in aanzienlijke mate beslag op schaarse ruimte; bij de huidige vraag naar ophoogzand zeker enige honderden hectaren per jaar.
- De vervangende ophoogmaterialen zullen een bescheiden rol blijven spelen; meer kan verwacht worden van partiëel ophogen en van hergebruik van ophoogzand.
- De overheid is doende beleid en wetgeving op te stellen voor de ontgrondingenproblematiek; met name de samenhang tussen het ontgrondingenbeleid en de ruimtelijke ordening moet hierbij beter gewaarborgd worden.
- De mogelijkheid bestaat dat het minimaliseringsprobleem, waaruit het kostenmodel bestaat, met een andere dan de onderzochte oplossingsmethoden kan worden opgelost. Of deze methode efficiënter is en voor grote problemen tot oplossing leidt is niet bekend.
- Met het standaard softwarepakket MPSX en met het computerprogramma dat op basis van het ICW-oplossingsalgorithme geschreven is, kan het minimaliseringsprobleem niet voor het volledige proefgebied worden opgelost.
- Wel kunnen met beide programma's deelproblemen worden opgelost, zij het dat het ICW-oplossingalgorithme niet consistent tot oplossing van alle kleinere problemen komt.
- De uitgangspunten van het kostenmodel zijn zodanig geformuleerd dat het model een redelijk waarheidsgetrouw beeld van de werkelijkheid geeft.
- Het kostenmodel is zodanig geformuleerd dat er in voldoende mate alternatief beleid mee kan worden afgewogen; winning vanuit zee kan echter niet afgewogen worden.
- Het kostenmodel is niet geschikt als bedrijfkundig instrument voor de particuliere ondernemer
- De aard van het model is zodanig, dat het een exact geformuleerde en efficiënte oplossingsmethode vereist.
- De provinciale overheden zijn organisatorisch in staat het kostenmodel daadwerkelijk te gebruiken, zonodig met externe ondersteuning.
- De kosten van het gebruik van het model blijven relatief beperkt.
- Hoewel de baten van het model pas bekend worden nadat het model

gedurende een zekere periode in gebruik is, lijkt gebruik van het model te rechtvaardigen.

Aanbevelingen voor verdere studie

- Aanbevolen wordt het knelpunt in het ICW-oplossingsalgorithme op te lossen, m.a.w. de onderlinge afhankelijkheid van de dummievariabelen in het oplossingsalgorithme te verwerken.
- Indien dit niet tot het gewenste resultaat leidt, kan het alternatieve oplossingsalgorithme, waarvan het principe in dit verslag is aangegeven, worden ontwikkeld.
- Hierbij wordt aanbevolen eerst een gedegen studie te doen naar bestaande software voor gemengd geheeltallige lineaire programmeringsproblemen; zowel naar algemene als meer specifieke software.
- Het verdient aanbeveling een bovengrens voor de putproduktie op te nemen.
- De invloed van de verschillende vereenvoudigingen in het model dient onderzocht te worden. Hier is echter een werkend oplossingsalgorithme voor nodig.

Slotconclusie

Geconcludeerd mag worden dat het kostenmodel een functie kan vervullen als provinciaal beleidsondersteunend instrument bij de verlening van ontgrondingsvergunningen, mits hiervoor een werkend oplossingsalgorithme beschikbaar komt. De baten van het gebruik van het model zijn hierbij niet zozeer gelegen in het bepalen van de minimale kosten van de ophoogzandwinning, maar in het kwantificeren van alternatief beleid, dat veelal door planologische overwegingen bepaald wordt.

Literatuuroverzicht

[1] ADREA, J.F.R., 1969. Vergelijking diverse transportmiddelen, Cursus Opspuiten Terreinen. Stichting Postacademisch Onderwijs in de Civiele Techniek, Delft
[2] HILLIER F.S. en LIEBERMAN G.J., 1974. Introduction to Operations Research. Holden-Day, San Francisco
[3] SEGEREN W.A. en HENGEVELD H., 1978. Bouwrijp maken van terreinen. College Polderinrichting F17, Afdeling der Civiele Techniek, Technische Hogeschool, Delft
[4] LOOTSMA F.A., 1978. Deterministic methods of Operations Research. Lecture Notes A87I, Department of Mathematics, University of Technology, Delft
[5] BAK R.L., 1979. Het ontgrondingenvraagstuk in Nederland. Kandidaatsscriptie Geografisch Planologisch Instituut, Vrije Universiteit, Amsterdam
[6] MEISSNER, L.P. en ORGANICK, E.I., 1979. FORTRAN 77 featuring structured programming. Addison-Wesley, New York.
[7] IJKELENSTAM, G.P.F., 1980. Plaatskeuze van zandwinobjecten, Deel 1. Technische en economische grondslagen. ICW-Nota 1168, Wageningen
[8] GOUDZWAARD, L. en 'T HOEN, H., 1980. Ophoogzandvoorziening van Noord-Holland. Afstudeerrapport Afdeling der Civiele Techniek, Technische Hogeschool, Delft
[9] VEEKE, H.P.M., 1981. Een MPSX-invoergenerator. Rapport 81-06, Onderafdeling Wiskunde en Informatica, Technische Hogeschool, Delft
[10] IJKELENSTAM G.P.F., 1981. Kosten van winning en transport van zand als criterium bij plaatsbepaling van zandwinobjecten. Polytechnisch tijdschrift 36 nr. 7.
[11] VAN LOHUIZEN, H.P.S., 1981. Ophoogzand. Civiele en Bouwkundige Techniek nr. 1, Doetinchem
[12] VREKE, J., 1982. Optimale allocatie van zandwinobjecten. ICW-Nota 1335, Wageningen
[13] VREKE, J., 1982. Benadering van de relatie tussen oppervlakte en inhoud van een zandwinput door een lineaire spline functie. ICW-Nota 1343, Wageningen
[14] MINISTERIE VAN VERKEER EN WATERSTAAT, 1983. UPD-Nota, uitgangspunten, probleemstelling en doelstellingen, langetermijn beleid voor de oppervlaktedelfstoffenvoorziening. 's-Gravenhage
[15] LANDELIJKE COMMISSIE VOOR DE COORDINATIE VAN HET ONTGRONDINGENBELEID, 1984. Nota Ophoogzand. 's-Gravenhage
[16] STICHTING STUDIECENTRUM WEGENBOUW, 1984. Zandwinning voor wegen: Waar en hoe?. Arnhem
[17] CENTRAAL BUREAU VOOR DE STATISTIEK, 1980/1986. Maandstatistiek voor de bouwnijverheid, Produktie van bouwmaterialen, tabel 10.1 M. Heerlen
[18] SANDROCK, K., 1988. A Simple Algorithm for Solving Small Fixed-Charge Transportation Problems. Journal of the Operational Research Society Vol. 39, No. 5, pp. 467-475. Great Britain
[19] IBM. Mathematical Programming System Extended/370 (MPSX/370), Program Reference Manual, SH19-1095-0
[20] IBM. Mathematical Programming System Extended/370 (MPSX/370), Mixed Integer Programming/370 (MIP/370), Program Reference Manual, SH19-1099-0

ICW Nota 1934
december 1988

EEN KOSTENMODEL VOOR DE OPHOOGZANDVOORZIENING

Bijlagen

Inhoud

I	Voorstel afstudeeronderzoek ICW en toelichting	1
II	Gegevens proefgebied	5
III	Toelichting op de branch-and-bound techniek	9
IV	Toelichting op het simplex algorithme voor transportproblemen	13
V	Het standaard simplex algorithme	21
VI	INPUT; listing, betekenis variabelen, in- en uitvoer deelprobleem	25
VII	Invoer GEN80 voor het deelprobleem	31
VIII	MPSX; opties, uitkomsten en handberekening deelprobleem	33
IX	MODIFY; listing en betekenis variabelen	37
X	MAIN; listing en betekenis variabelen	41
XI	RECALCO; listing en betekenis variabelen	45
XII	ANSOL; listing en betekenis variabelen	49
XIII	PRISOL; listing en betekenis variabelen	53
XIV	RESIM; listing en betekenis variabelen	61
XV	TRANS; listing en betekenis variabelen	67
XVI	CORRECT; listing en betekenis variabelen	75
XVII	ZEON; listing en betekenis variabelen	77
XVIII	SPLINE; listing, betekenis variabelen, in- en uitvoer rekenvoorbeeld	85
XIX	TESTRECA en RECALCO; listings, betekenis variabelen, in- en uitvoer test, controleberekening	89
XX	TESTANSOL en ANSOL; listings, betekenis variabelen, uitvoer test	99
XXI	TESTRESIM en RESIM; listings, betekenis variabelen, in- en uitvoer tests	109
XXII	TESTPRISOL en PRISOL; listings, betekenis variabelen, in- en uitvoer test, controleberekening	117
XXIII	TESTDUAL en CORRECT; listings, betekenis variabelen, in- en uitvoer test	131
XXIV	TESTTRANS en TRANS; listings, betekenis variabelen, in- en uitvoer test, controleberekening	141
XXV	TESTZEON en ZEON; listings, betekenis variabelen in- en uitvoer test	157
XXVI	MAIN; listing oorspronkelijke versie, uitvoer programmarun	163
XXVII	TESTMODIFY; listing, betekenis variabelen, in- en uitvoer test	171
XXVIII	In- en uitvoer bestanden van twee programmaruns met de laatste versie van het programma	179

BIJLAGE I

Het onderzoek met betrekking tot de allocatie van zandwinputten

Voor een ondersteuning van het provinciaal beleid met betrekking tot het verstrekken van ontgrondingsvergunningen is inzicht nodig in de invloed van het aantal, de omvang en de ligging van zandwinobjecten op de totale kosten. Het onderhavige onderzoek beoogt hieraan een bijdrage te leveren. Doel van het onderzoek is de formulering van een optimaliseringsmodel dat bij een gegeven vraag naar zand (zowel de gevraagde hoeveelheid als de plaats van bestemminggegeven) uit een aantal potentiële zandwinobjecten de uit het oogpunt van kosten minimalisering optimale keuze maakt. Dit houdt in dat wordt aangegeven:

- de zandwinobjecten die in produktie worden genomen;
- voor elk van de vraagpunten, de zandwinobjecten waaruit zand wordt verkregen en de wijze waarop het transport plaatsvindt. Onderscheid wordt gemaakt tussen transport per as en transport per persleiding;

Het onderzoek wordt uitgevoerd voor een probleem waarbij 19 potentiële zandwinobjecten en 47 bestemmingen werden onderscheiden.
Dit probleem is geformuleerd als een gemengd geheeltallig lineair programmeringsprobleem. Door de omvang van het probleem is, bij toepassing van een van de standaard oplossingsalgorithmes, een te groot aantal iteraties nodig om het optimum te bereiken. Door splitsing van het probleem in een normaal lineair programmeringsprobleem en een geheeltallig probleem, wordt geprobeerd het aantal iteraties te reduceren. Hiertoe is een optimaliseringsalgorithme geformuleerd.
De werkzaamheden die in het kader van het afstudeeronderwerp moeten worden verricht zijn:

- het schrijven van een computerprogramma voor het optimaliseringsalgorithme;
- het toetsen en indien nodig aanpassen van het optimaliseringsalgorithme;
- het toetsen van het optimaliseringsmodel op een probleem waarvan de parameters al zijn berekend.

Wageningen, maart 1982 drs.Jan Vreke

ONDERZOEK PLAATSKEUZE ZANDWINOBJECTEN

Sinds 1971 is 'de Ontgrondingenwet' van kracht waarbij iedere ontgronding gebonden is aan een vergunning, meestal af te geven door Gedeputeerde Staten.

Onder het thema 'Aspecten van zand- en grindvoorziening' werd op het 51e wegencongres in 1975 ruim aandacht geschonken aan de problematiek rond de winning van oppervlaktedelfstoffen waaronder de stagnatie bij de afgifte van vergunningen.

In 1976 wordt in opdracht van de Federatie van Oppervlaktedelfstoffenwinnende Industrieën (FODI) door de Stichting Het Nederlands Economisch Instituut (NEI) een onderzoek ingesteld naar het verwachte gebruik van oppervlaktedelfstoffen in Nederland.

Ter bevordering van een gecoördineerd beleid wordt in 1976 door de toenmalige minister van Verkeer en Waterstaat de Landelijke Commissie voor de Coördinatie van het Ontgrondingenbeleid (L.C.C.O.) ingesteld. Onder punt c van de taakomschrijving van de L.C.C.O. wordt aandacht geschonken aan de integratie van het ontgrondingenbeleid in de ruimtelijke ordening en de landinrichting (zie bijlage).

Min of meer tegelijkertijd wordt door de provincies een beleid ontwikkeld waarbij de voorkeur uitgaat naar centrale winplaatsen boven de tot dan veelal gebruikelijke zandput zo dicht mogelijk bij de bestemming. Een en ander versterkt de behoefte aan onderzoek ten behoeve van een meer afgewogen keuze van winplaatsen.

In 1977 doet IJkelenstam het voorstel, in verband met de vele raakvlakken van de problematiek met de landinrichting, een onderzoek in te stellen naar de optimale plaatskeuze van zandwinobjecten. Hij is thans als projectleider verantwoordelijk voor het betreffende onderzoek.

Het probleemveld blijkt nog nauwelijks onderzocht, de L.C.C.O. toont belangstelling (secr. destijds ir. L.A. van Duyvendijk). Analytisch onderzoek wordt uitgevoerd hetgeen de noodzakelijke technische en economische grondslagen legt voor de uit te voeren optimalisering (IJkelenstam, 1980). Bestaande optimaliseringsmodellen zoals toegepast ten behoeve van de locatie van diverse voorzieningen (bijvoorbeeld vuilverwerking, grasdrogerijen of andere bedrijven) blijken ongeschikt voor het complexe probleem. De afdeling Algemene Economie van het ICW wordt gevraagd een model

te formuleren dat binnen de opzet van het onderzoek wel tot bevredigende resultaten leidt. De formulering van zo'n model dat mede is gebaseerd op gemengd geheeltallig lineaire programmering is inmiddels ver gevorderd.

De complexiteit van het probleem en de behoefte aan onderzoek op dit terrein wordt nog eens onderstreept in een recent artikel van prof.ir. H.P. van Lohuizen (zie bijlage). Onder zijn leiding is een soortgelijk onderzoek verricht ten behoeve van Noord-Holland in het kader van het afstuderen aan de TH Delft door L. Goudswaard en H. 't Hoen. In hun onderzoek is transport per persleiding niet als alternatief opgenomen. Transport per persleiding als alternatief heeft nu juist tot gevolg dat de bestaande modellen niet bruikbaar zijn.

Bij transport per auto of schip wordt gebruik gemaakt van bestaande auto- of vaarwegen en heeft de korrelverdeling van het zand geen invloed op de kosten. Bij hydraulisch transport daarentegen moet het tracé worden aangelegd en is het aantal te plaatsen tussenstations afhankelijk van de gewenste perssnelheid welke per korrelgrootte verschilt. Daarnaast moeten de kosten van verschillende tracé's over meerdere bestemmingen verdeeld worden. Deze verschillen leiden tot een meer ingewikkelde modelstructuur met veel variabelen. Bij binnendijkse centrale zandwinning ligt toepassing van hydraulisch transport uit een oogpunt van kostenminimalisatie het meest voor de hand.

Ideevorming ten aanzien van centrale winplaatsen welke worden gekozen op basis van afweging van vele belangen, is betrekkelijk kort in ontwikkeling en ook binnen de landinrichting nieuw. Hiernaar wordt door Rijks- en Provinciale diensten zelf geen vergelijkbaar onderzoek verricht. Wel vindt veelvuldig overleg plaats in diverse ambtelijke werkgroepen, ook interdepartementaal, waarbij het accent ligt op de formulering van criteria ten behoeve van het ontgrondingenbeleid. Incidenteel wordt daartoe in opdracht door derden onderzoek verricht. In dit verband kan worden genoemd het onderzoek dat in opdracht van de Provincie Limburg wordt uitgevoerd door het Planologisch Studiecentrum-TNO te Delft. Het onderzoek dat nog niet is afgerond betreft een evaluatie van potentiële grindwinplaatsen.

Uitgaande van diverse belangen, samenhangend met onder meer natuurbehoud of grindwinning wordt een multicriteria-analyse uitgevoerd. (projectleider is ir. A. van Setten van TNO). Bij grindwinning is trans-

port per schip met aansluitende distributie via de zand- en grindhandel gebruikelijk. Aan hydraulisch grondverzet, aan de orde bij transport van zand, wordt daarbij anders dan in het ICW onderzoek geen aandacht geschonken. Introductie van deze vervoerswijze maakt het optimaliseringsprobleem complexer.

Hydraulisch grondverzet als belangrijk facet van het ICW onderzoek bepaalt dan ook in belangrijke mate het eigen karakter van dit onderzoek.

LITERATUUR

L.C.C.O. Taakomschrijving

Lohuizen, H.P.S. van, 1981. Ophoogzand
Civiele en bouwkundige techniek nr. 1

IJkelenstam, G.F.P., 1977. Onderzoeksvoorstel afweging zandwinning
Notitie ICW

IJkelenstam, G.F.P., 1977. Afweging locatie zandwinprojecten
Notite ICW

IJkelenstam, G.F.P., 1980. Plaatskeuze van zandwinobjecten deel 1.
Technische en economische grondslagen
ICW nota 1168

IJkelenstam, G.F.P., 1981. Kosten van winning en transport van zand als criterium bij plaatsbepaling van zandwinobjecten.
Pol. techn. tijdschr. 36 nr. 7

Wageningen, 19-2-1982
ing. G.F.P. IJkelenstam

Gegevens proefgebied

Het proefgebied is gelegen in de Betuwe ten oosten van Dodewaard. De provincie Gelderland heeft voor de periode 1977/1986 de behoefte aan ophoogzand per gemeente in dit proefgebied geraamd. Hiermee is de vraag naar ophoogzand en de geografische verdeling van de vraagpunten over deze periode bekend. Het aantal vraagpunten bedraagt 47.

Naast 5 bestaande winplaatsen zijn 14 potentiële winplaatsen aangewezen op grond van bodemkundige en geologische kenmerken. Het proefgebied omvat dus in totaal 19 zandwinlocaties.

In fig. 1 van het verslag is het proefgebied met de vraagpunten, de bestaande en de potentiële winplaatsen weergegeven. Voor Nijmegen wordt aangenomen dat het daar benodigde ophoogzand ten zuiden van de Waal gewonnen wordt, zodat de vraag naar ophoogzand van Nijmegen niet is opgenomen.

De omvang van de vraag van de 47 vraagpunten is weergegeven in tabel A.

vraagpunt	vraag	vraagpunt	vraag	vraagpunt	vraag	vraagpunt	vraag
1	15	13	200	25	40	37	40
2	65	14	500	26	40	38	40
3	350	15	75	27	40	39	40
4	50	16	75	28	40	40	40
5	10	17	75	29	40	41	40
6	30	18	75	30	40	42	40
7	150	19	75	31	40	43	40
8	1000	20	75	32	40	44	40
9	200	21	75	33	40	45	40
10	1000	22	75	34	40	46	40
11	500	23	40	35	40	47	40
12	1000	24	40	36	40		

Tabel A Gevraagde hoeveelheid zand per vraagpunt [1000 m^3] (nummering van de vraagpunten volgens fig. 1)

Voor het kostenmodel zijn de volgende gegevens van de winputten van belang:

- Het talud van winning [-].
- De grondprijs.
- De vaste aanlegkosten.
- De vaste winningskosten bij transport per pijpleiding.
- De vaste winningskosten bij transport per vrachtwagen.
- De korreldiameter van het zand [μ].
- De dikte van de niet uit zand bestaande bovenlaag [m].
- De dikte van de winbare zandlaag [m].

Het talud van winning is voor alle winputten gesteld op 1:4, de grondprijs op f 50.000,-/ha, de vaste aanlegkosten op f 100.000,-, de vaste winningskosten bij transport per pijpleiding op f 17.000,- en de vaste winningskosten bij transport per vrachtwagen op f 32.000,-. De overige gegevens van de winputten zijn opgenomen in tabel B.

winput	korreldiam. [μ]	dikte bovenlaag [m]	dikte zandlaag [m]
1	400	0,4	19,6
2	700	4,0	19,0
3	500	2,0	21,0
4	500	2,0	28,5
5	400	2,0	23,0
6	200	3,7	17,8
7	150	3,5	9,0
8	400	0,4	19,6
9	600	3,3	26,7
10	500	4,2	18,8
11	300	5,5	23,5
12	500	1,0	24,0
13	500	1,8	28,2
14	200	1,5	15,5
15	800	5,6	16,4
16	400	2,0	18,0
17	400	2,2	10,0
18	800	2,5	30,0
19	800	1,6	19,9

Tabel B Gegevens van de winputten (nummering van de winputten volgens fig. 1)

De invloed van de korreldiameter is in het kostenmodel gekoppeld aan de coëfficiënten α en δ. De relatie tussen de korreldiameters en deze coëfficiënten is weergegeven in tabel C. Opgemerkt moet worden dat deze relatie niet specifiek is voor het proefgebied, doch in het algemeen geldt.

In het kostenmodel wordt onderscheid gemaakt tussen wegtransport en transport per pijpleiding. Voor de transportkosten van het transport per pijpleiding is naast de korreldiameter van het getransporteerde zand de lengte van het leidingtracé van belang. In fig. 14 van het verslag is het leidingennet van het proefgebied weergegeven, dat alle winputten met alle vraagpunten verbindt. In tabel D zijn de lengten van de pijpleidingtrajecten opgenomen.

Voor de transportkosten van het wegtransport zijn de transportafstanden tussen de winputten en de vraagpunten van belang. In tabel E zijn de gewogen transportafstanden tussen de winputten en de vraagpunten opgenomen.

korreldiameter [μ]	α	δ
150	26000	0,1585
200	29100	0,2080
300	34550	0,2615
400	39350	0,2985
500	43850	0,3355
600	48300	0,3730
700	52500	0,3985
800	56600	0,4150

Tabel C Relatie tussen de korreldiameters en de coëfficiënten α en δ

traject	lengte	traject	lengte	traject	lengte	traject	lengte
1	36	23	10	45	9	67	11
2	27	24	33	46	11	68	10
3	26	25	19	47	21	69	12
4	9	26	38	48	43	70	10
5	7	27	24	49	9	71	5
6	16	28	50	50	4	72	10
7	5	29	12	51	44	73	14
8	10	30	15	52	18	74	19
9	7	31	62	53	15	75	15
10	5	32	38	54	19	76	6
11	11	33	15	55	9	77	6
12	10	34	20	56	12	78	11
13	13	35	9	57	11	79	10
14	10	36	29	58	11	80	13
15	11	37	23	59	10	81	11
16	9	38	8	60	11	82	10
17	7	39	39	61	8	83	7
18	12	40	16	62	2	84	9
19	7	41	34	63	10	85	12
20	15	42	23	64	7	86	11
21	7	43	45	65	2		
22	9	44	11	66	4		

Tabel D Lengte van de pijpleidingtrajecten [hm] (nummering trajecten volgens fig. 14)

vraagpunt	winput																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	12	9	16	21	26	6	9	12	14	9	9	12	12	14	19	17	21	22	20
2	2	10	11	15	16	7	4	1	3	10	13	12	15	15	6	12	11	12	16
3	11	6	3	7	11	11	13	9	7	5	9	2	7	6	6	4	8	9	7
4	17	10	4	3	6	18	18	15	12	11	8	8	7	6	13	3	12	9	2
5	10	2	6	10	15	9	9	8	6	2	6	3	7	6	9	8	11	12	10
6	24	16	10	7	10	23	23	21	18	19	19	14	11	11	18	9	15	14	5
7	17	16	9	9	2	20	20	14	15	15	19	12	16	13	9	8	5	3	7
8	13	14	6	10	8	14	16	12	10	10	14	7	10	9	7	7	4	5	10
9	13	13	8	12	5	16	15	11	11	13	17	8	13	12	6	8	3	1	8
10	13	14	6	10	8	14	16	12	10	10	14	7	10	9	7	7	4	5	10
11	20	17	11	15	4	20	18	15	16	15	19	12	8	15	10	11	6	3	8
12	18	12	7	11	6	16	16	12	11	11	15	8	16	11	7	8	4	2	7
13	14	14	9	13	4	18	16	12	13	13	17	10	14	13	8	9	4	2	8
14	20	17	11	15	4	20	18	15	16	15	19	12	16	15	10	11	6	3	8
15	8	7	9	14	16	10	7	3	1	7	11	4	13	11	8	10	13	15	10
16	12	11	13	18	19	14	11	7	4	11	15	8	17	15	8	14	13	15	14
17	11	13	8	13	14	17	10	7	8	10	14	6	12	11	4	9	8	11	18
18	7	9	6	11	13	13	9	6	7	9	13	5	11	10	3	7	7	10	13
19	15	10	6	11	13	14	8	7	8	10	14	6	11	10	2	7	8	11	11
20	16	14	10	15	15	18	9	10	11	14	18	10	15	14	2	11	12	9	15
21	16	14	9	15	11	16	16	12	11	12	16	11	12	11	7	10	4	5	15
22	19	10	5	10	7	15	16	11	10	10	14	7	11	9	6	6	3	4	10
23	12	8	15	21	24	5	8	11	13	8	8	11	11	13	18	16	20	21	19
24	13	8	14	20	23	6	9	12	14	8	7	11	10	12	17	14	19	20	18
25	12	8	14	18	23	7	10	11	14	6	6	10	9	11	16	13	18	19	17
26	11	7	12	17	22	8	9	11	13	5	5	9	8	10	15	11	17	18	16
27	10	6	11	16	21	7	8	10	12	4	4	8	7	9	14	10	16	17	15
28	10	4	10	15	20	8	9	11	10	3	3	7	6	8	13	9	16	16	14
29	11	4	11	13	20	9	10	12	11	4	2	8	5	7	14	10	17	18	13
30	12	3	11	12	18	10	11	13	11	5	1	9	4	6	14	10	16	17	11
31	14	2	10	11	16	12	13	13	10	4	2	8	3	5	13	9	15	16	10
32	15	3	8	10	14	13	14	14	11	5	3	9	2	4	14	7	13	14	9
33	16	4	7	9	13	14	15	15	12	6	4	8	1	3	13	6	12	13	8
34	17	5	8	8	13	15	16	15	13	7	5	9	1	2	13	6	13	13	8
35	18	6	9	7	14	16	17	16	14	9	7	10	1	3	14	7	14	14	9
36	19	8	9	5	12	18	19	17	14	10	8	10	2	4	15	17	14	15	7
37	20	10	9	3	11	20	21	17	15	11	10	11	4	6	15	16	14	13	6
38	22	12	8	2	11	21	22	18	14	13	11	11	5	7	15	6	13	12	6
39	23	13	6	1	11	22	23	18	15	14	12	11	6	8	15	7	13	12	5
40	25	14	7	2	11	23	24	19	16	15	13	12	7	9	15	8	13	12	4
41	26	15	8	3	11	24	25	19	15	16	14	11	8	10	15	7	13	11	3
42	27	17	9	5	11	26	27	20	16	17	16	12	9	11	16	8	13	12	4
43	28	18	10	6	11	27	28	21	17	19	17	13	10	12	17	9	14	13	5
44	29	19	10	7	10	28	29	22	18	20	18	14	11	13	18	9	15	14	5
45	31	20	11	8	9	29	30	23	19	21	19	15	12	14	19	10	16	15	6
46	32	22	12	9	9	30	31	24	20	22	20	16	13	15	20	11	17	16	7
47	33	23	13	10	10	31	32	25	21	23	21	17	14	16	21	12	18	17	8

Tabel E Gewogen rijafstanden [km] tussen winputten en vraagpunten (nummering winputten en vraagpunten volgens fig. 1)

BIJLAGE III

Toelichting op de branch-and-bound techniek

Beschouw het volgende gemengd geheeltallige lineaire minimaliseringsprobleem:

$$\min x_0 = \sum_{i=1}^{n} (c_i x_i) + \sum_{j=1}^{m} (c_j d_j) \qquad (1)$$

$$\text{ST} \sum_{i=1}^{n} (a_{ki} x_i) + \sum_{j=1}^{m} (a_{kj} d_j) \geq b_k; \quad k = 1,.., p \qquad (2)$$

$$x_i \geq 0; \; i = 1,..., n, \qquad (3)$$

$$d_j = 0 \text{ of } 1; \; j = 1,..., m \qquad (4)$$

In dit probleem is er sprake van continue (x_i) en discrete (0-1) variabelen (d_j).

Het oplossen van dit probleem m.b.v. de branch-and-bound techniek verloopt volgens de volgende stappen:

1. Vervang de beperkingen (4) door de beperkingen (5) en (6):

   $$d_j \geq 0; \; j = 1,..., m \qquad (5)$$

   $$d_j \leq 1; \; j = 1,..., m \qquad (6)$$

   en los het probleem, dat hiermee volledig continue geworden is, op m.b.v. het standaard simplex algorithme.

   De oplossing kan naast de continue x_i en de niet-geheeltallige d_j $(0 < d_j < 1)$ een aantal d_j bevatten die 0 of 1 zijn.

   Stel nu:

   $x_0 = M_0$, de waarde van het in stap 1 bereikte minimum

   $J0^- = \{j \mid 0 < d_j < 1\}$

   $J0^+ = \{j \mid d_j = 0, 1\}$

   De waarde van de optimale oplossing van het gemengd geheeltallige probleem zal nooit kleiner zijn dan de waarde van de continue oplossing M_0; M_0 is de absolute ondergrens van de waarde van de uiteindelijke oplossing.

2. Kies die d_{j1}; $j1 \in J0^-$ waarvan verwacht wordt dat het effect op de minimalisering het grootst is.

Los nu, uitgaande van de in stap 1 bereikte oplossing twee nieuwe continue lineaire minimaliseringsproblemen op m.b.v. het standaard simplex algorithme; één met $d_{j1} = 0$ en één met $d_{j1} = 1$. Hiertoe worden de beperkingen (5) en (6) vervangen door de beperkingen (7) t/m (9) respectievelijk (10) t/m (12):

$$d_j \geq 0;\ j \in J0^-;\ j \neq j1 \tag{7}$$

$$d_j \leq 1;\ j \in J0^-;\ j \neq j1 \tag{8}$$

$$d_{j1} = 0 \tag{9}$$

en

$$d_j \geq 0;\ j \in J0^-;\ j \neq j1 \tag{10}$$

$$d_j \leq 1;\ j \in J0^-;\ j \neq j1 \tag{11}$$

$$d_{j1} = 1 \tag{12}$$

Als uitkomsten van deze twee lineaire minimaliseringsproblemen zal men de minima $x_0 = M_{10}$ en $x_0 = M_{11}$ vinden (voor $d_{j1} = 0$ respectievelijk $d_{j1} = 1$). Deze minima zijn altijd groter dan het absolute minimum M_0. $M_1 = \min\{M_{10}, M_{11}\}$ is nu de nieuwe ondergrens voor de optimale oplossing.

Stap 2 wordt herhaald totdat alle d_j de waarde 0 of 1 gekregen hebben. Voor de duidelijkheid wordt dit weergegeven in stap 3 (de 2e vertakking van de branch-and-bound techniek).

3. $J1^- = \{j \mid 0 < d_j < 1\}$

$J1^+ = \{j \mid d_j = 0, 1\}$

en kies de d_{j2}; $j2 \in J1^-$ waarvan verwacht wordt dat het effect op de minimalisering het grootst is.

Voeg nu de vergelijkingen (13) t/m (15) respectievelijk (16) t/m (18) toe aan het stelsel (7) t/m (9) of (10) t/m (12) afhankelijk van de d_{j1} waarvoor geldt $M_1 = \min\{M_{10}, M_{11}\}$.

$$d_j \geq 0;\ j \in J1^-;\ j \neq j2 \tag{13}$$

$$d_j \leq 1;\ j \in J1^-;\ j \neq j2 \tag{14}$$

$$d_{j2} = 0 \tag{15}$$

en

$$d_j \geq 0;\ j \in J1^-;\ j \neq j2 \tag{16}$$

$$d_j \leq 1;\ j \in J1^-;\ j \neq j2 \tag{17}$$

$$d_{j2} = 1 \tag{18}$$

Het resultaat is twee nieuwe lineaire minimaliseringsproblemen die m.b.v. het standaard simplex algorithme kunnen worden opgelost.

De structuur van de herhalingen kan worden geïllustreerd middels de beslissingsboom in fig. A.

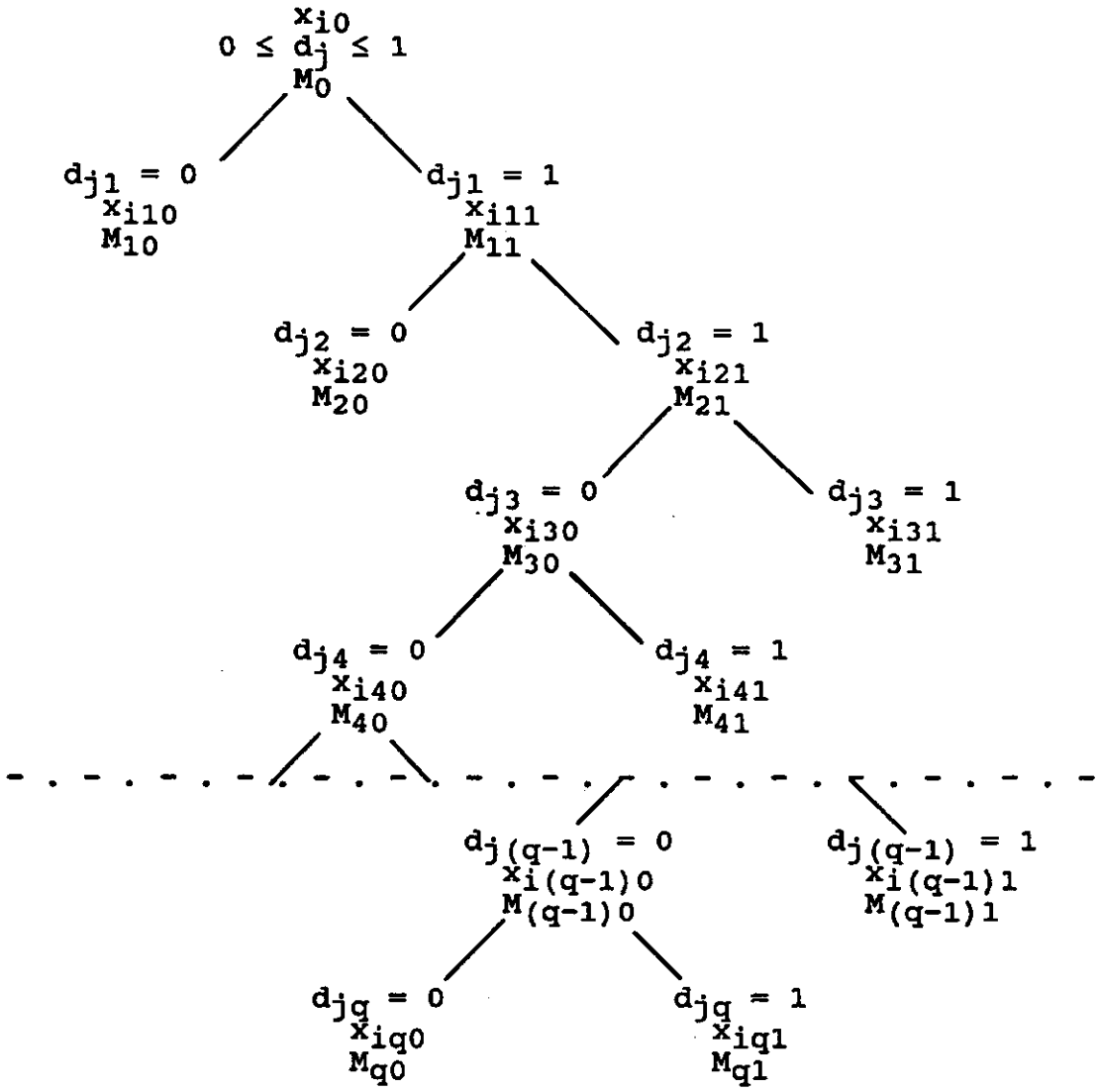


Fig. A Structuur van de branch-and-bound techniek

De uiteindelijk gevonden oplossing $M_q = \min\{M_{q0}, M_{q1}\}$ is niet zonder meer de optimale oplossing. Voor de takken die afgebroken zijn omdat het minimum M_r van de tak voor d_{jr} groter was dan voor $1 - d_{jr}$ ($d_{jr} = 0$ of 1) moet gecontroleerd worden of dit minimum M_r niet kleiner is dan M_q. Zo ja, dan moeten deze takken gevolgd worden, m.a.w. dan moet voor ieder van deze takken de stappen 2 en 3 herhaald worden, totdat er een $x_0 = M$ gevonden wordt die groter is dan M_q en de tak definitief kan worden afgesloten. De optimale oplossing kan zich dus in een tak bevinden die in eerste instantie was afgebroken.

BIJLAGE IV

Toelichting op het simplex algorithme voor transportproblemen

Het simplex algorithme voor transportproblemen wordt toegelicht middels het volgende transportprobleem, dat is ontleend aan [4].

$$
\begin{aligned}
\min\ & 2x_{11} + 3x_{12} + 11x_{13} + 7x_{14} + \\
& x_{21} + 6x_{23} + x_{24} + \\
& 5x_{31} + 8x_{32} + 15x_{33} + 9x_{34}
\end{aligned}
$$

$$
\begin{aligned}
\text{ST}\quad & x_{11} + x_{12} + x_{13} + x_{14} = 6 \\
& x_{21} + x_{22} + x_{23} + x_{24} = 1 \\
& x_{31} + x_{32} + x_{33} + x_{34} = 10 \\
& x_{11} + x_{21} + x_{31} = 7 \\
& x_{12} + x_{22} + x_{32} = 5 \\
& x_{13} + x_{23} + x_{33} = 3 \\
& x_{14} + x_{24} + x_{34} = 2
\end{aligned}
\tag{19}
$$

met $x_{ij} \geq 0$; $i = 1, \ldots, 3$ en $j = 1, \ldots, 4$

Oftewel:

$$\min \sum_{i=1}^{3} \left(\sum_{j=1}^{4} (c_{ij}x_{ij}) \right)$$

$$\text{ST}\quad \sum_{j=1}^{4} (x_{ij}) = S_i;\ i = 1, \ldots, 3$$

$$\sum_{i=1}^{3} (x_{ij}) = D_j;\ j = 1, \ldots, 4$$

$$x_{ij} \geq 0;\ i = 1, \ldots, 3;\ j = 1, \ldots, 4$$

met:

$$c_{ij} = \begin{bmatrix} 2 & 3 & 11 & 7 \\ 1 & & 6 & 1 \\ 5 & 8 & 15 & 9 \end{bmatrix} \qquad S_i = \begin{bmatrix} 6 \\ 1 \\ 10 \end{bmatrix} \qquad D_i = \begin{bmatrix} 7 \\ 5 \\ 3 \\ 2 \end{bmatrix}$$

Het probleem (19) voldoet aan de kenmerken van een transportprobleem:

1. De coëfficiënten van de variabelen in de beperkingen hebben de waarde 0 of 1.
2. De vraag is gelijk aan het aanbod: $\sum_{i=1}^{m} (S_i) = \sum_{j=1}^{n} (D_j)$
   $(6 + 1 + 10 = 7 + 5 + 3 + 2 = 17)$

In fig. B wordt probleem (19) in schemavorm weergegeven. De horizontale rijen geven de aanbodbeperkingen weer en de verticale kolommen de vraagbeperkingen. In de vakken zijn de waarden van de transportvariabelen x_{ij} opgenomen. De kostencoëfficiënten c_{ij} zijn in de aparte vakjes linksboven in de vakken voor de x_{ij} opgenomen.

Het oplossen van dit transportprobleem volgens de simplex methode voor transportproblemen verloopt volgens de volgende stappen:

1. Bepaal de startbasisvariabelen. Hiervoor bestaan verschillende methoden. Een efficiënte methode is de regel van Vogel ([2] en [4]):

   - Benut de goedkoopste transportroute optimaal: $c_{22} = 0$ en $x_{22} = 1$.
   - Rij 2 is nu uitgeput; benut nu de goedkoopste transportroute uit kolom 2: $c_{12} = 3$ en $x_{12} = 4$.
   - Kolom 2 is nu eveneens uitgeput; benut nu de goedkoopste transportroute uit rij 1: $c_{11} = 2$ en $x_{11} = 2$.
   - Analoog aan het voorgaande wordt gevonden: $x_{31} = 5$, $x_{34} = 2$ en $x_{33} = 3$.
   - De waarde van de bijbehorende oplossing is 104 (1*0 + 4*3 + 2*2 + 5*5 + 2*9 + 3*15 = 104).

   Deze startoplossing voor de basisvariabelen is weergegeven in fig. B.

	naar 1	2	3	4	aanbod
van 1	2 / 2	3 / 4	11 / -	7 / -	6
van 2	1 / -	0 / 1	6 / -	1 / -	1
van 3	5 / 5	8 / -	15 / 3	9 / 2	10
vraag	7	5	3	2	min=104

Fig. B Probleem (19) in schemavorm met de startoplossing voor de basisvariabelen volgens de regel van Vogel

2. Bepaal de getransformeerde kosten. Bepaal hiertoe eerst de simplex multipliers v_i en w_j. Dit zijn de duale variabelen die corresponderen met de aanbod- en de vraagbeperkingen.

Bij de bepaling van de basisvariabelen m.b.v. de regel van Vogel is gebruik gemaakt van de beperkingen (20):

$$\begin{aligned} x_{22} &= 1 \\ x_{22} + x_{12} &= 5 \\ x_{12} + x_{11} &= 6 \\ x_{11} + x_{31} &= 7 \\ x_{34} &= 2 \\ x_{33} &= 3 \end{aligned} \qquad (20)$$

Voor de simplex multipliers die corresponderen met deze beperkingen (20) waaruit de basisvariabelen zijn opgelost geldt:

$$v_i + w_j = c_{ij};\ i = 1,\ldots, 3;\ j = 1,\ldots, 4$$

De simplex multipliers kunnen nu worden opgelost uit het stelsel (21):

$$\begin{aligned} v_2 + w_2 &= 0 \\ v_1 + w_2 &= 3 \\ v_1 + w_1 &= 2 \\ v_3 + w_1 &= 5 \\ v_3 + w_4 &= 9 \\ v_3 + w_3 &= 15 \end{aligned} \qquad (21)$$

Het stelsel (21) bestaat uit 6 vergelijkingen en 7 variabelen en kan dus worden opgelost door één variabele arbitrair een waarde te geven. Kies hiervoor de variabele die in de meeste vergelijkingen voorkomt (v_3) en geef hem de waarde 0. De uitkomsten voor v_i en w_j worden dan:

$$\begin{aligned} v_3 &= 0 \\ w1 &= 5 \\ w4 &= 9 \\ w3 &= 15 \\ v1 &= 2 - w1 = -3 \\ w2 &= 3 - v1 = 6 \\ v2 &= 0 - w2 = -6 \end{aligned} \qquad (22)$$

Met behulp van deze simplex multipliers kunnen de getransformeerde kosten worden bepaald. Voor de basisvariabelen zijn deze gelijk aan 0 ($v_i + w_j - c_{ij} = 0$) en voor de niet basisvariabelen gelijk aan:

$$v_i + w_j - c_{ij} \qquad (23)$$

M.b.v. van vergelijking (23) en de simplex multipliers (22) kunnen de getransformeerde kosten worden berekend; de uitkomsten zijn weergegeven in fig. C.

4. Zolang er nog positieve getransformeerde kosten zijn, kan de oplossing verbeterd worden. Kies de grootste getransformeerde kosten (waarde 3) en neem de bijbehorende transportvariabele in de basis op (x_{23}).

   Bepaal nu de maximale waarde van x_{23}, bepaal welke variabele de basis moet verlaten en bereken de waarden van de andere basisvariabelen. In fig. D is dit weergegeven: x_{23} krijgt de waarde θ en de consequenties voor de andere basisvariabelen worden bepaald. Omdat alle $x_{ij} \geq 0$ moeten zijn kan θ maximaal de waarde 1 krijgen ($1 - \theta \geq 0 \Rightarrow \theta \leq 1 \Rightarrow \theta = 1$). Dit resulteert in de volgende waarden voor de basisvariabelen:

$x_{23} = 1$ (wordt in de basis opgenomen)	$x_{11} = 2 - 1 = 1$
$x_{22} = 1 - 1 = 0$ (verlaat de basis)	$x_{12} = 4 + 1 = 5$
$x_{31} = 5 + 1 = 6$	$x_{33} = 3 - 1 = 2$
$x_{34} = 2$ (onveranderd)	

In de fig. E t/m H worden de stappen 3 en 4 herhaald. Uit fig. G blijkt dat er in de derde simplex iteratie geen positieve getransformeerde kosten meer zijn, zodat de optimale oplossing bereikt is.

	naar 1	2	3	4	
van 1	2 / -	3 / -	11 / 1	7 / -1	$v_1=-3$
2	1 / -2	0 / -	6 / 3	1 / 2	$v_2=-6$
3	5 / -	8 / -2	15 / -	9 / -	$v_3= 0$
	$w_1=5$	$w_2=6$	$w_3=15$	$w_4=9$	

Fig. C Getransformeerde kosten in de eerste simplex iteratie

	naar 1	2	3	4	aanbod
van 1	2 / $2-\theta$	3 / $4+\theta$	11 / -	7 / -	6
2	1 / -	0 / $1-\theta$	6 / $+\theta$	1 / -	1
3	5 / $5+\theta$	8 / -	15 / $3-\theta$	9 / 2	10
vraag	7	5	3	2	$\theta=1$ min=101

Fig. D De eerste simplex iteratie; x_{23} wordt in de basis opgenomen en x_{22} verlaat de basis

	naar 1	2	3	4	
van 1	2 / -	3 / -	11 / 1	7 / -1	$v_1=-3$
van 2	1 / -5	0 / -3	6 / -	1 / -1	$v_2=-9$
van 3	5 / -	8 / -2	15 / -	9 / -	$v_3= 0$
	$w_1=5$	$w_2=6$	$w_3=15$	$w_4=9$	

Fig. E Getransformeerde kosten in de tweede simplex iteratie

	naar 1	2	3	4	aanbod
van 1	2 / 1-θ	3 / 5	11 / +θ	7 / -	6
van 2	1 / -	0 / -	6 / 1	1 / -	1
van 3	5 / 6+θ	8 / -	15 / 2-θ	9 / 2	10
vraag	7	5	3	2	θ=1 min=100

Fig. F De tweede simplex iteratie; x_{13} wordt in de basis opgenomen en x_{11} verlaat de basis

	naar 1	2	3	4	
van 1	2 / -1	3 / -	11 / -	7 / -2	$v_1=-4$
2	1 / -5	0 / -2	6 / -	1 / -1	$v_2=-9$
3	5 / -	8 / -1	15 / -	9 / -	$v_3= 0$
	$w_1=5$	$w_2=7$	$w_3=15$	$w_4=9$	

Fig. G Getransformeerde kosten in de derde simplex iteratie

	naar 1	2	3	4	aanbod
van 1	2 / -	3 / 5	11 / 1	7 / -	6
2	1 / -	0 / -	6 / 1	1 / -	1
3	5 / 7	8 / -	15 / 1	9 / 2	10
vraag	7	5	3	2	min=100

Fig. H De derde simplex iteratie; het optimum is bereikt

Het standaard simplex algorithme

In bijlage IV is het transportprobleem (19) opgelost met het simplex algorithme voor transportproblemen:

$$
\begin{array}{lllllll}
\min & 2x_{11} & + 3x_{12} & + 11x_{13} & + 7x_{14} & + & \\
 & x_{21} & & + 6x_{23} & + x_{24} & + & \\
 & 5x_{31} & + 8x_{32} & + 15x_{33} & + 9x_{34} & & \\
\end{array}
$$

$$
\begin{array}{lllllll}
\text{ST} & x_{11} + x_{12} + x_{13} + x_{14} & = & 6 & & \\
 & x_{21} + x_{22} + x_{23} + x_{24} & = & 1 & & \\
 & x_{31} + x_{32} + x_{33} + x_{34} & = & 10 & & (19) \\
 & x_{11} + x_{21} + x_{31} & = & 7 & & \\
 & x_{12} + x_{22} + x_{32} & = & 5 & & \\
 & x_{13} + x_{23} + x_{33} & = & 3 & & \\
 & x_{14} + x_{24} + x_{34} & = & 2 & &
\end{array}
$$

met $x_{ij} \geq 0$; $i = 1, \ldots, 3$ en $j = 1, \ldots, 4$

Bij het oplossen van dit probleem m.b.v. het standaard simplex algorithme moet een speciale methode gehanteerd worden om startwaarden voor de basisvariabelen te vinden. Dit is nodig omdat de beperkingen gelijkheden zijn en er dus geen natuurlijke spelingsvariabelen kunnen worden ingevoerd, die als startbasisvariabelen kunnen dienen.

Eem methode die met name voor computerberekening zeer geschikt is, is de zgn. Phase I procedure ([2] en [4]). Hierbij wordt aan iedere beperking een kunstmatige spelingsvariabelen y_i toegevoegd ($i = 1, \ldots, n$, met n: totaal aantal beperkingen). Deze kunstmatige variabelen worden vervolgens geëlimineerd door de doelstellingsfunctie $y_0 = \sum_{i=1}^{n} \{y_i\}$ te minimaliseren.

In fig. I is het stelsel vergelijkingen met de kunstmatige variabelen y_i weergegeven. Het stelsel vergelijkingen waarmee het standaard simplex algorithme start, wordt verkregen door de y_i uit de doelstellingsfunctie te elimineren door alle beperkingen erbij op te tellen. Dit is weergegeven in fig. J. Het bijbehorende starttableau is weergegeven in fig. K. Het simplex tableau waarin alle y_i geëlimineerd zijn levert de startbasis voor het standaard simplex algorithme. Uitgaande van dit tableau kan x_0 geminimaliseerd worden.

Bij vergelijking van het starttableau K van het standaardsimplex algorithme met de tableau's C t/m H van het simplex algorithme voor transportproblemen blijkt duidelijk dat het tableau voor het standaard simplex algorithme aanzienlijk groter is.

$$\min y_0$$

$$
\begin{array}{l}
y_0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad -y_1-y_2-y_3-y_4-y_5-y_6-y_7 = 0 \\
x_0-2x_{11}-3x_{12}-11x_{13}-7x_{14}-x_{21} \quad -6x_{23}-x_{24}-5x_{31}-8x_{32}-15x_{33}-9x_{34} \qquad\qquad\qquad\qquad = 0 \\
\quad x_{11}+x_{12}+x_{13}+x_{14} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad +y_1 \qquad\qquad\qquad\qquad = 6 \\
\qquad\qquad\qquad\qquad x_{21}+x_{22}+x_{23}+x_{24} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad +y_2 \qquad\qquad\qquad = 1 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad x_{31}+x_{32}+x_{33}+x_{34} \qquad\qquad\qquad\qquad +y_3 \qquad\qquad = 10 \\
\quad x_{11} \qquad\qquad\qquad +x_{21} \qquad\qquad\qquad\qquad + x_{31} \qquad\qquad\qquad\qquad\qquad\qquad +y_4 \qquad\qquad = 7 \\
\qquad x_{12} \qquad\qquad\qquad\qquad +x_{22} \qquad\qquad\qquad\qquad + x_{32} \qquad\qquad\qquad\qquad\qquad\qquad +y_5 \qquad = 5 \\
\qquad\qquad x_{13} \qquad\qquad\qquad\qquad + x_{23} \qquad\qquad\qquad\qquad + x_{33} \qquad\qquad\qquad\qquad\qquad\qquad +y_6 \quad = 3 \\
\qquad\qquad\qquad x_{14} \qquad\qquad\qquad\qquad +x_{24} \qquad\qquad\qquad\qquad + x_{34} \qquad\qquad\qquad\qquad\qquad\qquad +y_7 = 2
\end{array}
$$

$x_{ij} \geq 0;\ i = 1, \ldots, 3$ en $j = 1, \ldots, 4$
$y_k \geq 0;\ k = 1, \ldots, 7$

Fig. I Stelsel vergelijkingen bij gebruik van de Phase I procedure

$$\min y_0$$

$$
\begin{array}{l}
y_0+2x_{11}+2x_{12}+2x_{13}+2x_{14}+2x_{21}+2x_{22}+2x_{23}+2x_{24}+2x_{31}+2x_{32}+2x_{33}+2x_{34} \qquad\qquad\qquad\qquad = 34 \\
x_0-2x_{11}-3x_{12}-11x_{13}-7x_{14}-x_{21}- \quad 6x_{23}-x_{24}-5x_{31}-8x_{32}-15x_{33}-9x_{34} \qquad\qquad\qquad\qquad = 0 \\
\quad x_{11}+x_{12}+x_{13}+x_{14} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad +y_1 \qquad\qquad\qquad\qquad = 6 \\
\qquad\qquad\qquad\qquad x_{21}+x_{22}+x_{23}+x_{24} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad +y_2 \qquad\qquad\qquad = 1 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad x_{31}+x_{32}+x_{33}+x_{34} \qquad\qquad\qquad\qquad +y_3 \qquad\qquad = 10 \\
\quad x_{11} \qquad\qquad\qquad + x_{21} \qquad\qquad\qquad\qquad + x_{31} \qquad\qquad\qquad\qquad\qquad\qquad +y_4 \qquad\qquad = 7 \\
\qquad x_{12} \qquad\qquad\qquad\qquad + x_{22} \qquad\qquad\qquad\qquad + x_{32} \qquad\qquad\qquad\qquad\qquad\qquad +y_5 \qquad = 5 \\
\qquad\qquad x_{13} \qquad\qquad\qquad\qquad + x_{23} \qquad\qquad\qquad\qquad + x_{33} \qquad\qquad\qquad\qquad\qquad\qquad +y_6 \quad = 3 \\
\qquad\qquad\qquad x_{14} \qquad\qquad\qquad\qquad + x_{24} \qquad\qquad\qquad\qquad + x_{34} \qquad\qquad\qquad\qquad\qquad\qquad +y_7 = 2
\end{array}
$$

$x_{ij} \geq 0;\ i = 1, \ldots, 3$ en $j = 1, \ldots, 4$
$y_k \geq 0;\ k = 1, \ldots, 7$

Fig. J Stelsel vergelijkingen waarmee het standaard simplex algorithme start

basis	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	x_{23}	x_{24}	x_{31}	x_{32}	x_{33}	x_{34}	y_1	y_2	y_3	y_4	y_5	y_6	y_7	rhs
y_0	2	2	2	2	2	2	2	2	2	2	2	2								34
x_0	-2	-3	-11	-7	-1		6	-1	-5	-8	-15	-9								0
	1	1	1	1									1							6
					1	1	1	1						1						1
									1	1	1	1			1					10
	1				1				1							1				7
		1				1				1							1			5
			1				1				1							1		3
				1				1				1							1	2

Fig. K starttableau voor de Phase I procedure

BIJLAGE VI

Listing van het programma INPUT

```
      PROGRAM INPUT
      INTEGER KU,M,N,SU
      PARAMETER(KU=5,M=5,N=2,SU=25)
      INTEGER A(M),ALFA(N),B(N),CA(N),CB(N),COEF(SU,M,N),CW(N),DT(SU),
     #        DW(N,M),I,J,K,QA,QBU(N,SU),S
      REAL AU(N),D(N),DB(N,M),DELTA(N),E(N,SU),F(N,M),G(N,M),H(N,KU),
     #     KQ(N,KU),PG(N),Q(N,KU),R(N,0:KU),RA(N,0:KU),RO,RQ(N,KU),
          RQ0(N),T(N),V(N)
      DATA RO /0.01/
      READ(5,*) COEF
      DO 10 I=1,N
            READ(5,*) ALFA(I),CA(I),CB(I),CW(I),D(I),DELTA(I),
     #                PG(I),RQ0(I),T(I),V(I),(DW(I,J),J=1,M)
   10 CONTINUE
      READ (5,*) A,DT
      QA=0
      DO 20 J=1,M
            QA=QA+A(J)
   20 CONTINUE
      DO 110 I=1,N
            R(I,0)=(2*T(I)*V(I)+(6*T(I)*RQ0(I))**(1.0/3))**2
            RA(I,0)=4*T(I)*(6*T(I)*RQ0(I))**(-2.0/3)*(2*T(I)*V(I)+
     #              (6*T(I)*RQ0(I))**(1.0/3))
            AU(I)=PG(I)*(R(I,0)-RQ0(I)*RA(I,0))+CA(I)+CB(I)
            B(I)=CW(I)-CB(I)
            DO 40 S=1,SU
                  DO 30 J=1,M
                        IF(COEF(S,J,I).EQ.0)GOTO 30
                        E(I,S)=ALFA(I)*DT(S)/10.0
                        GOTO 40
   30             CONTINUE
                  E(I,S)=0.0
   40       CONTINUE
            DO 60 J=1,M
                  F(I,J)=PG(I)*RA(I,0)+2.6+0.35*DW(I,J)
                  DB(I,J)=0.0
                  DO 50 S=1,SU
                        DB(I,J)=DB(I,J))+COEF(S,J,I)*DT(S)/10.0
   50             CONTINUE
                  G(I,J)=PG(I)*RA(I,0)+0.66+DELTA(I)*DB(I,J)
   60       CONTINUE
            RQ(I,1)=4.0/3*D(I)**3*T(I)**2
            DO 70 K=1,KU
                  R(I,K)=((RQ(I,K)/D(I)-T(I)**2*D(I)**2/3)**0.5+T(I)*
     #                  D(I)+2*T(I)*V(I))**2
                  RA(I,K)=(1+(T(I)*D(I)+2*T(I)*V(I))/(RQ(I,K)/D(I)-
     #                   T(I)**2*D(I)**2/3)**0.5)/D(I)
                  H(I,K)=PG(I)*(RA(I,K)-RA(I,K-1))
                  IF(K.EQ.1)THEN
                    KQ(I,1)=(R(I,1)-R(I,0)+RQ0(I)*RA(I,0)-RQ(I,1)*
     #                     RA(I,1))/(RA(I,0)-RA(I,1))
```

```
                ENDIF
                Q(I,K)=QA-KQ(I,K)
                IF(K.EQ.KU)GOTO 80
                KQ(I,K+1)=T(I)**2*D(I)**3/3+D(I)*(((R(I,K)*(D(I)*
   #                     RA(I,K)-1)*RO)**0.5+(1+RO)*(T(I)*D(I)+2*
   #                     T(I)*V(I)))/(D(I)*RA(I,K)-1-RO))**2
                RQ(I,K+1)=T(I)**2*D(I)**3/3+D(I)*(KQ(I,K+1)/D(I)-
   #                     T(I)**2*D(I)**2/3)**2*(RQ(I,K)/D(I)-
   #                     T(I)**2*D(I)**2/3)
 70         CONTINUE
 80         CONTINUE
            DO 100 S=1,SU
                  QBU(I,S)=0
                  DO 90 J=1,M
                        QBU(I,S)=QBU(I,S)+COEF(S,J,I)*A(J)
 90               CONTINUE
100         CONTINUE
110 CONTINUE
    WRITE(3,5) A
    WRITE(3,15) (RQ0(I),I=,N)
    WRITE(3,25) QA
    WRITE(3,35) COEF
    WRITE(3,45) QBU
    WRITE(3,55) KQ
    WRITE(3,65) Q
    WRITE(3,75) AU
    WRITE(3,85) B
    WRITE(3,95) E
    WRITE(3,105) F
    WRITE(3,115) G
    WRITE(3,125) H
    WRITE(3,135)
    STOP
  5 FORMAT('A(J)',6X,':',X,5(X,I11)/(12X,5(X,I11)))
 15 FORMAT('RQ0(I)',4X,':',X,5(X,F11.1)/(12X,5(X,F11.1)))
 25 FORMAT('QA',8X,':',2X,I11)
 35 FORMAT('COF(I,J,S):',X,25(X,I1)/(12X,25(X,I1)))
 45 FORMAT('QBU(S,I)',2X,':',X,5(X,I11)/(12X,5(X,I11)))
 55 FORMAT('KQ(K,I)',3X,':',X,5(X,F11.1)/(12X,5(X,F11.1)))
 65 FORMAT('Q(K,I)',4X,':',X,5(X,F11.1)/(12X,5(X,F11.1)))
 75 FORMAT('AU(I)',5X,':',X,5(X,F11.1)/(12X,5(X,F11.1)))
 85 FORMAT('B(I)',6X,':',X,5(X,I11)/(12X,5(X,I11)))
 95 FORMAT('E(S,I)',4X,':',X,5(X,F11.1)/(12X,5(X,F11.1)))
105 FORMAT('F(J,I)',4X,':',X,5(X,F11.3)/(12X,5(X,F11.3)))
115 FORMAT('G(J,I)',4X,':',X,5(X,F11.3)/(12X,5(X,F11.3)))
125 FORMAT('H(K,I)',4X,':',X,5(X,F11.3)/(12X,5(X,F11.3)))
135 FORMAT('''END''')
    END
```

Betekenis van de in het programma INPUT gebruikte variabelen

A(J) : vraag naar zand van vraagpunt J [m^3]
AU(I) : parameter voor de vaste kosten van het in produktie nemen van winput I [f]

ALFA(I)	:	kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand van winput I [f/km]
B(I)	:	parameter voor de vaste kosten van het transport per vrachtwagen vanuit winput I [f]
CA(I)	:	vaste kosten van het in produktie nemen van winput I [f]
CB(I)	:	vaste winningskosten van winput I bij transport per pijpleiding [f]
COEF(S,J,I)	:	0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
CW(I)	:	vaste winningskosten van winput I bij transport per vrachtwagen [f]
D(I)	:	dikte van de laag winbaar zand van winput I [m]
DB(I,J)	:	lengte van het totale leidingtracé dat winput I met vraagpunt J verbindt [km]
DELTA(I)	:	kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand van winput I [f/km]
DT(S)	:	lengte van leidingtraject S [hm]
DW(I,J)	:	gewogen transportafstand van winput I naar vraagpunt J bij transport per vrachtwagen [km]
E(I,S)	:	parameter voor de vaste kosten van traject S van het leidingennet van winput I [f]
F(I,J)	:	coëfficiënt voor de variabele kosten van het transport per vrachtwagen van winput I naar vraagpunt J [f/m^3]
G(I,J)	:	coëfficiënt voor de variabele kosten van het transport per pijpleiding van winput I naar vraagpunt J [f/m^3]
H(I,K)	:	coëfficiënt voor de reductie in variabele kosten wanneer de produktie van winput I een hoeveelheid KQ(I,K) overschrijdt [f/m^3]
I	:	index van de winputten [-]
J	:	index van de vraagpunten [-]
K	:	index van de spline functie intervallen [-]
KQ(I,K)	:	zandproduktie van winput I waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KU	:	aantal spline functie intervallen [-]
M	:	aantal vraagpunten [-]
N	:	aantal winputten [-]
PG(I)	:	grondprijs van winput I [f/m^2]
Q(I,K)	:	QA - KQ(I,K) [m^3]
QA	:	totale vraag naar zand [m^3]
QBU(I,S)	:	maximale hoeveelheid zand die vanuit winput I getransporteerd wordt door leidingtraject S [m^3]
R(I,K)	:	terreinoppervlakte als functie van de zandproduktie RQ(I,K) van winput I [m^2]
RA(I,K)	:	afgeleide van de functie R(I,K) [m^{-1}]
RO	:	relatieve afwijking $\frac{RS(q)-R(q)}{R(q)} < RO$ [-]
RQ(I,K)	:	zandproduktie van winput I die correspondeert met het begin van het K-de spline functie interval [m^3]

RQ0(I) : minimaal te produceren hoeveelheid zand ingeval winput I in produktie genomen wordt [m^3]
S : index van de leidingtrajecten [-]
SU : aantal leidingtrajecten [-]
T(I) : talud van de winput I [-]
V(I) : dikte van de niet uit zand bestaande bovenlaag van winput I [m]

Invoergegevens van het deelprobleem

Het aantal pijpleidingtrajecten is handmatig teruggebracht van 86 tot 25 (SU=25). De trajecten, die geen deel uitmaken van het deelnetwerk dat de winputten 4 en 18 verbindt met de vraagpunten 8, 30, 39, 40 en 41 zijn komen te vervallen en de nummering van de overblijvende trajecten is aangepast (zie fig. 21 van het verslag)

```
KU=5 M=5 N=2 SU=25
Winputten 4 18: I=1-2
Vraagpunten 8 30 39 40 41: J=1-5
Trajecten 18 26 27 40 41 44-48 53-55 67-70 72-79: S=1-25
RO          = 0,01
COEF(S,J,I)= 1 1 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1
             1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
             0 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
             0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
             0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
             0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
ALFA(I)     = 43850  56600
CA(I)       = 100000  100000
CB(I)       = 17000  17000
CW(I)       = 32000  32000
D(I)        = 28,5  30,0
DELTA(I)    = 0,3355  0,4150
PG(I)       = 5,0  5,0
RQ0(I)      = 50000,0  50000,0
T(I)        = 4,0  4,0
V(I)        = 2,0  2,5
DW(I,J)     = 10 12  1  2  3
               5 17 12 12 11
A(J)        = 1000000  40000  40000  400000  40000
DT(S)       = 12 38 24 16 34 11  9 11 21 43
              15 19  9 11 10 12 10 10 14 19
              15  6  6 11 10
```

Invoerbestand van het deelprobleem

```
2*1,4*0,3*1,0,2*1,13*0,
12*0,10*1,3*0,
12*0,1,9*0,1,2*0,
```

```
12*0,1,9*0,2*1,0,
12*0,1,9*0,3*1,
1,5*0,2*1,17*0,
3*0,3*1,3*0,1,3*0,3*1,9*0,
0,2*1,5*0,1,14*0,1,0,
0,2*1,5*0,1,16*0,
0,2*1,5*0,1,15*0,1,
43850,100000,17000,32000,28.5,0.3355,5.0,50000.0,4.0,2.0,
10,12, 1, 2, 3,
56600,100000,17000,32000,30.0,0.4150,5.0,50000.0,4.0,2.5,
 5,17,12,12,11,
1000000,4*40000,
12,38,24,16,34,11, 9,11,21,43,15,19, 9,11,10,12,10,10,14,19,
15, 6, 6,11,10
```

Uitvoerbestand van het deelprobleem

```
A(J)        :       1000000        40000        40000        40000
40000
RQ0(I)      :      50000.0      50000.0
QA          :      1160000
COF(I,J,S):  1 1 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1
             1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
             0 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
             0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
             0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
             0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
QBU(S,I)    :      1000000      1000000      1000000       120000            0
                    120000            0        40000            0        40000
                         0        40000      1000000      1000000      1000000
                   1000000      1000000       120000            0        40000
                   1000000            0      1000000            0       160000
                         0        40000        40000        40000        40000
                     40000        40000        40000            0        40000
                         0        40000            0        40000            0
                     40000            0        40000            0       120000
                         0        80000        40000        40000        40000
KQ(K,I)     :     164678.6     178120.9     675775.7     788107.9    1365155.0
                 1590295.0    3030738.0    3518880.0    7536637.0    8694603.0
Q(K,I)      :     995321.4     981879.1     484224.3     371892.1    -205155.0
                 -430295.0   -1870738.0   -2358880.0   -6376637.0   -7534603.0
AU(I)       :     148435.7     151989.5
B(I)        :        15000        15000
E(S,I)      :      52620.0      67920.0     166630.0     215080.0          0.0
                  135840.0          0.0      90560.0          0.0     192440.0
                       0.0      62260.0      39465.0      50940.0      48235.0
                   62260.0      92085.0     118860.0          0.0     243380.0
                   65775.0          0.0      83315.0          0.0      39465.0
                       0.0      48235.0      62260.0      43850.0      56600.0
                   52620.0      67920.0      43850.0          0.0      43850.0
                       0.0      61390.0          0.0      83315.0          0.0
```

		65775.0	0.0	26310.0	0.0	26310.0
		0.0	48235.0	62260.0	43850.0	56600.0
F(J,I)	:	6.966	5.245	7.666	9.445	3.816
		7.695	4.166	7.695	4.516	7.345
G(J,I)		5.720	2.883	5.418	7.240	2.029
		5.456	2.398	4.999	2.734	5.414
H(K,I)	:	-0.491	-0.553	-0.066	-0.064	-0.045
		-0.044	-0.032	-0.031	-0.022	-0.022

'END'

BIJLAGE VII

Invoer GEN80 voor het deelprobleem

```
'MODEL'SAND
'INDICES'
I 1:2
J 1:5
S 1:25
K 1:5
'RESTRICTIONS'
R1(J)   :SIGMA_I(QB(I,J)+QW(I,J))=A(J)
R2(I)   :SIGMA_J(QB(I,J)+QW(I,J))-RQ0(I)*D1(I)>=0
R3(I)   :SIGMA_J(QB(I,J)+QW(I,J))-QA*D1(I)<=0
R4(I)   :SIGMA_J(QW(I,J))-QA*D2(I)<=0
R5(I,S):SIGMA_J(COF(I,J,S)*QB(I,J))-QBU(S,I)*D3(I,S)<=0
R6(I,K):SIGMA_J(QB(I,J)+QW(I,J))-QX(I,K)-KQ(K,I)*D4(I,K)>=0
R7(I,K):QX(I,K)-Q(K,I)*D4(I,K)<=0
'OBJ'
COST:SIGMA_I(AU(I)*D1(I))+SIGMA_I(B(I)*D2(I))+
SIGMA_I(SIGMA_S(E(S,I)*D3(I,S)))+SIGMA_I(SIGMA_J(F(J,I)*QW(I,J)))+
SIGMA_I(SIGMA_J(G(J,I)*QB(I,J)))+SIGMA_I(SIGMA_K(H(K,I)*QX(I,K)))
'INTEGER'
D1,D2,D3,D4
'DATA'
A(J)        :      1000000       40000       40000       40000       40000
RQ0(I)      :      50000.0     50000.0
QA          :      1160000
COF(I,J,S):  1 1 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0
             0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1
             1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
             0 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0
             0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
             0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
             0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
QBU(S,I)    :      1000000     1000000     1000000      120000           0
                    120000           0       40000           0       40000
                         0       40000     1000000     1000000     1000000
                   1000000     1000000      120000           0       40000
                   1000000           0     1000000           0      160000
                         0       40000       40000       40000       40000
                     40000       40000       40000           0       40000
                         0       40000           0       40000           0
                     40000           0       40000           0      120000
                         0       80000       40000       40000       40000
KQ(K,I)     :     164678.6    178120.9    675775.7    788107.9   1365155.0
                 1590295.0   3030738.0   3518880.0   7536637.0   8694603.0
Q(K,I)      :     995321.4    981879.1    484224.3    371892.1   -205155.0
                -430295.0  -1870738.0  -2358880.0  -6376637.0  -7534603.0
AU(I)       :     148435.7    151989.5
B(I)        :        15000       15000
```

E(S,I)	:	52620.0	67920.0	166630.0	215080.0	0.0
		135840.0	0.0	90560.0	0.0	192440.0
		0.0	62260.0	39465.0	50940.0	48235.0
		62260.0	92085.0	118860.0	0.0	243380.0
		65775.0	0.0	83315.0	0.0	39465.0
		0.0	48235.0	62260.0	43850.0	56600.0
		52620.0	67920.0	43850.0	0.0	43850.0
		0.0	61390.0	0.0	83315.0	0.0
		65775.0	0.0	26310.0	0.0	26310.0
		0.0	48235.0	62260.0	43850.0	56600.0
F(J,I)	:	6.966	5.245	7.666	9.445	3.816
		7.695	4.166	7.695	4.516	7.345
G(J,I)		5.720	2.883	5.418	7.240	2.029
		5.456	2.398	4.999	2.734	5.414
H(K,I)	:	-0.491	-0.553	-0.066	-0.064	-0.045
		-0.044	-0.032	-0.031	-0.022	-0.022

'END'

Opties waarmee MPSX gedraaid is

```
PROGRAM
INITIALZ
MOVE(XDATA,'SAND')
MOVE(XPBNAME,'F.A.VAN REST')
MOVE(XOBJ,'COST')
MOVE(XRHS,'RHSIDE')
CONVERT('SUMMARY')
BCDOUT
SETUP('MIN')
OPTIMIZE
SOLUTION
OPTIMIX
EXIT
PEND
```

Uitkomsten van MPSX voor het deelprobleem

```
Continue oplossing        : 3398913,7065
Optimale integer oplossing: 3678862,5453
d1(1): 1, d1(2): 1
d2(1): 1, d2(2): 0
d3(1,13), d3(1,23) - d3(1,25): 1, overige d3(1,s): 0
d3(2,1), d3(2,7), d3(2,8)    : 1, overige d3(2,s): 0
d4(1,1) - d4(1,5): 0
d4(2,1), d4(2,2) : 1, overige d4(2,k): 0
qw(1,2): 40000.0, qw(1,5): 40000.0, overige qw(1,j): 0
qw(2,1) - qw(2,5): 0
qb(1,3): 40000.0, qb(1,4): 40000.0, overige qb(1,j): 0
qb(2,1): 1000000.0, overige qb(2,j): 0
```

D.w.z. transport per vrachtwagen vanuit winput 1 naar de vraagpunten 2 en 5 en transport per pijpleiding vanuit winput 1 naar de vraagpunten 3 en 4 en vanuit winput 2 naar vraagpunt 1.

Handberekening voor het deelprobleem

Op basis van de handmatig berekende transportkosten zijn voorspellingen t.a.v. de keuze van de transportmethode gedaan. Hierbij is gebruik gemaakt van de gewogen transportafstanden en de totale lengte van de leidingtracé's (tabel F).

<u>Vraagpunt 1</u>

Vanuit winput 1:

CTW=15000+1000000*(2,6+0,35*10)= f 6.115.000,-
CTB=43850*12,5+1000000*(0,66+0,3355*12,5)= f 5.402.875,-

Vanuit winput 2:

CTW=15000+1000000*(2,6+0,35*5)= f 4.365.000,-
CTB=56600*3,2+1000000*(0,66+0,4150*3,2)= f 2.169.120,-

Vraagpunt 2

Vanuit winput 1:

CTW=15000+40000*(2,6+0,35*12)= f 287.000,-
CTB=43850*11,6+40000*(0,66+0,3355*11,6)= f 690.732,-

Vanuit winput 2:

CTW=15000+40000*(2,6+0,35*17)= f 357.000,-
CTB=56600*13,7+40000*(0,66+0,4150*13,7)= f 1.029.240,-

Vraagpunt 3

Vanuit winput 1:

CTW=15000+40000*(2,6+0,35*1)= f 133.000,-
CTB=43850*1,5+40000*(0,66+0,3355*1,5)= f 112.305,-

Vanuit winput 2:

CTW=15000+40000*(2,6+0,35*12)= f 287.000,-
CTB=56600*9,4+40000*(0,66+0,4150*9,4)= f 714.480,-

Vraagpunt 4

Vanuit winput 1:

CTW=15000+40000*(2,6+0,35*2)= f 147.000,-
CTB=43850*2,6+40000*(0,66+0,3355*2,6)= f 175.302,-

Vanuit winput 2:

CTW=15000+40000*(2,6+0,35*12)= f 287.000,-
CTB=56600*8,3+40000*(0,66+0,4150*8,3)= f 633.960,-

Vraagpunt 5

Vanuit winput 1:

CTW=15000+40000*(2,6+0,35*3)= f 161.000,-
CTB=43850*3,6+40000*(0,66+0,3355*3,6)= f 232.572,-

Vanuit winput 2:

CTW=15000+40000*(2,6+0,35*11)= f 273.000,-
CTB=56600*9,3+40000*(0,66+0,4150*9,3)= f 707.160,-

Deze berekening wordt samengevat in tabel G.

vraagpunt	1		2		3		4		5	
	dw(1)	ΣD(s)	dw(2)	ΣD(s)	dw(3)	ΣD(s)	dw(4)	ΣD(s)	dw(5)	ΣD(s)
winput 1	10	12,5	12	11,6	1	1,5	2	2,6	3	3,6
winput 2	5	3,2	17	13,7	12	9,4	12	8,3	11	9,3

Tabel F Gewogen transportafstanden en totale lengte van de leidingtracé's [km]

winput	1		2	
transport	vrachtwagen	pijpleiding	vrachtwagen	pijpleiding
vraagpunt				
1	6.115.000	5.402.875	4.365.000	2.169.120
2	287.000	690.732	357.000	1.029.240
3	133.000	112.305	287.000	714.480
4	147.000	175.302	287.000	633.960
5	161.000	232.572	273.000	707.160

Tabel G Samenvating van de handmatig berekende transportkosten [f]

Op grond van deze uitkomsten zal het transport alsvolgt plaatsvinden:

Transport per vrachtwagen vanuit winput 1 naar de vraagpunten 2, 4 en 5 en transport per pijpleiding vanuit winput 1 naar vraagpunt 3 en vanuit winput 2 naar vraagpunt 1.

Deze uitkomsten komen voor het vraagpunt 4 niet overeen met de MPSX-uitkomsten. Echter, bij de handberekening zijn per vraagpunt voor het transport per pijpleiding de vaste kosten van het gehele traject in rekening gebracht. Omdat voor vraagpunt 3 transport per pijpleiding vanuit winput 1 het goedkoopste is worden de pijpleidingtrajecten 13 en 23 aangelegd. De vaste kosten van deze pijpleidingtrajecten hoeven nu niet meer meegerekend te worden voor de kosten van transport per pijpleiding vanuit winput 1 naar het vraagpunt 4.

De kosten wijzigen zich alsvolgt:

<u>Vraagpunt 4</u>

Vanuit winput 1:

CTB= 175.302-1,5*43.850= f 109.527,-

Hierdoor is transport per pijpleiding vanuit winput 1 voor vraagpunt 4 goedkoper dan transport per vrachtwagen.

In deze handberekening is niet gekeken naar de invloed van de aankoopkosten van de terreinen en naar de vaste aanlegkosten. Men kan zich nu afvragen of het in produktie nemen van een winput niet meer kosten met zich meebrengt dan de transportkosten die worden bespaard. De totale kosten bedragen volgens MPSX f 3.678.863,-. Het aandeel van de transportkosten hierin is volgens de handberekening f 2.796.514,-. Het aandeel van de aankoopkosten van de terreinen, de aanlegkosten en de vaste winningskosten bedraagt dus voor de twee winputten f 3.678.863,- - f 2.796.514,-= f 882.349,-. Door winput 2 in produktie te nemen wordt een bedrag van f 5.402.875,- - f 2.169.120,- = f 3.233.755,- op de transportkosten bespaard. Winput 2 wordt dus hoe dan ook in produktie genomen worden. Door winput 1 in produktie te nemen wordt een bedrag op de transportkosten bespaard van (f 357.000,- - f 287.00,-) + (f 287.000,-- f 112.305,-) + (f 287.000,- - f 109.527,-) + (f 273.000,- - f 118.562,-) = f 576.606,-

Nu moet bepaald worden welke kosten gemaakt worden wanneer winput 1 in produktie genomen wordt, met een produktie van $q = 160.000\ m^3$:

- De aankoopkosten van het terrein:

  $rq(1) = \frac{4}{3}*(4,0)^2(28,5)^3 = 493.848\ m^3 \geq 160.000\ m^3$

  dus:

  $R(q) = \{2*4,0*0,4 + (6*4,0*160.000)^{1/3}\}^2 = 25.534,3\ m^2$

  CAK = 5*25.534,3 = f 127.671,-

  (hierbij is vanwege de eenvoud gerekend met de werkelijke relatie tussen benodigd terreinoppervlak en putproduktie en niet met de lineaire spline functie benadering)

- De vaste aanleg- en winningskosten:

  CA = f 100.000,-, CW = f 32.000,- en CB = f 15.000,-

De totale kosten die gemaakt worden wanneer winput 1 in produktie genomen wordt bedragen f 127.671,- + f 100.000,- + f 32.000,-- f 15.000,- = f 244.671,-, hetgeen lager is dan de besparing op de transportkosten bij het in produktie nemen van winput 1, zijnde f 576.606,- (in werkelijkheid is het nog gunstiger om winput 1 in produktie te nemen omdat de aankoopkosten van het terrein van winput 2 lager zullen zijn wanneer winput 1 in produktie wordt genomen).

De uitkomsten van de handberekening komen dus overeen met de uitkomsten van MPSX voor dit deelprobleem.

BIJLAGE IX

Listing van het programma MODIFY

```
      PROGRAM MODIFY
C
C     Declareren van de parameters
C
      INTEGER CO,KU,M,N,SU
C
C     Definiëren van de parameters
C
      PARAMETER (KU=5,M=5,N=2,SU=86,CO=2*N+N*(SU+KU))
C
C     Declareren van de variabelen
C
      INTEGER COEF(N,SU,M),COMB(CO,0:25),COMBMAX,DT(N,SU),DTO(SU),
     #        END(N,M),G(SU,0:M),ND,SI(N),SIMAX,SIVAR1,SIVAR2
C
C     Declareren van de indices
C
      INTEGER I,J,JD,JH,JH1,JH2,K,S,SH,SH1
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*63 AREA
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) AREA
      WRITE(2,25) DATE,AREA
C
C     Inlezen van de invoergegevens
C
      READ(1,*) (((COEF(I,S,J),S=1,SU),J=1,M),I=1,N)
      READ(1,*) ((END(I,J),I=1,N),J=1,M)
      READ(1,*) (DTO(S),S=1,SU)
C
      JD=0
      SIMAX=0
      COMBMAX=0
      DO 220 I=1,N
             SI(I)=0
C
C     Bepalen van de afzonderlijke leidingnetten per winput
C
             DO 30 S=1,SU
                   DO 20 J=1,M
                         IF(COEF(I,S,J).EQ.1)THEN
                           SI(I)=SI(I)+1
                           SH=SI(I)
                           DT(I,SH)=DTO(S)
                           DO 10 JH=1,M
```

```
                                    COEF(I,SH,JH)=COEF(I,S,JH)
                                    IF(END(I,JH).EQ.S)END(I,JH)=SH
   10                         CONTINUE
                              GOTO 30
                            ENDIF
   20                 CONTINUE
   30           CONTINUE
C
C     Berekenen van G(S,0) en G(S,JH)
C
                DO 50 S=1,SI(I)
                      G(S,0)=0
                      DO 40 J=1,M
                            IF(COEF(I,S,J).EQ.1)THEN
                              G(S,0)=G(S,0)+1
                              JH=G(S,0)
                              G(S,JH)=J
                            ENDIF
   40                 CONTINUE
   50           CONTINUE
C
C     Samenvoegen van trajecten
C
                SIVAR1=SI(I)
                SIVAR2=SI(I)
                DO 130 S=1,SI(I)-1
                       DO 120 SH=SIVAR1,S+1,-1
                              DO 60 JH=0,G(S,0)
                                    IF(G(S,JH).NE.G(SH,JH))GOTO 120
   60                         CONTINUE
                              DT(I,S)=DT(I,S)+DT(I,SH)
                              DO 70 JH=1,G(SH,0)
                                    JH1=G(SH,JH)
                                    IF(END(I,JH1).EQ.SH)END(I,JH1)=S
   70                         CONTINUE
                              DO 110 SH1=SH,SIVAR2-1
                                     DT(I,SH1)=DT(I,SH1+1)
                                     DO 80 J=1,M
                                           COEF(I,SH1,J)=COEF(I,SH1+1,J)
   80                                CONTINUE
                                     DO 90 JH=1,G(SH1+1,0)
                                           JH1=G(SH+1,JH)
                                           IF(END(I,JH1).EQ.SH1+1)
     #                                       END(I,JH1)=SH1
   90                                CONTINUE
                                     DO 100 JH=0,G(SH1+1,0)
                                            G(SH1,JH)=G(SH1+1,JH)
  100                                CONTINUE
  110                         CONTINUE
                              SIVAR2=SIVAR2-1
  120                 CONTINUE
                      SIVAR1=SIVAR2
  130           CONTINUE
C
C     Bepalen van de combinatiematrix
C
```

```
            SI(I)=SIVAR2
            COMB(JD+1,0)=0
            DO 140 J=JD+2,JD+2+SI(I)+KU
                   COMB(J,0)=1
                   COMB(J,1)=JD+1
140         CONTINUE
            DO 170 S=1,SI(I)
                   J=JD+2+S
                   DO 160 SH=1,SI(I)
                          IF(SH.EQ.S)GOTO 160
                          DO 150 JH1=1,G(S,0)
                                 JH2=G(S,JH1)
                                 IF(COEF(I,SH,JH2).EQ.0)GOTO 160
150                       CONTINUE
                          COMB(J,0)=COMB(J,0)+1
                          JH=COMB(J,0)
                          COMB(J,JH)=JD+2+SH
160                CONTINUE
170         CONTINUE
            DO 190 K=1,KU
                   J=JD+2+SI(I)+K
                   DO 180 KH=1,KU
                          IF(KH.EQ.K)GOTO 190
                          COMB(J,0)=COMB(J,0)+1
                          JH=COMB(J,0)
                          COMB(J,JH)=JD+2+SI(I)+KH
180                CONTINUE
190         CONTINUE
            IF(SI(I).GT.SIMAX)SIMAX=SI(I)
C
C     Uitvoeren van de berekende variabelen
C
            WRITE(2,35) SI(I)
            DO 200 J=1,M
                   WRITE(2,45) (COEF(I,S,J),S=1,SI(I))
200         CONTINUE
            WRITE(2,55) (DT(I,S),S=1,SI(I))
            WRITE(2,35) (END(I,J),J=1,M)
            WRITE(2,35) (COMB(J,0),J=JD+1,JD+2+SI(I)+KU)
            DO 210 J=JD+2,JD+2+SI(I)+KU
                   WRITE(2,55) (COMB(J,JH),JH=1,COMB(J,0))
                   IF(COMB(J,0).GT.COMBMAX)COMBMAX=COMB(J,0)
210         CONTINUE
            JD=JD+2+SI(I)+KU
220 CONTINUE
    ND=JD
    WRITE(2,65) ND,SIMAX,COMBMAX
    STOP
  5 FORMAT(A10)
 15 FORMAT(A63)
 25 FORMAT(X,A/X,A)
 35 FORMAT(X,25I3)
 45 FORMAT(X,75I1)
 55 FORMAT(X,15I4)
 65 FORMAT(/X,'ND:',X,I4,3X,'SIMAX:'X,I2,3X,'COMBMAX:',X,I2)
    END
```

Betekenis van de in het programma MODIFY gebruikte variabelen

AREA	: identificatie van het proefgebied [-]
CO	: aantal dummievariabelen volgens de oorspronkelijke nummering [-]
COEF(I,S,J)	: 0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
COMB(J,0)	: aantal dummievariabelen dat afhankelijk is van dummievariabele X1(J) [-]
COMB(J,JH)	: combinatiematrix die aangeeft welke dummievariabelen afhankelijk zijn van dummievariabele X1(J) [-]
COMBMAX	: maximum aantal afhankelijke dummievariabelen [-].
DATE	: datum van de programma-run [-]
DT(I,S)	: lengte van traject S van het leidingennet van winput I [hm]
DTO(S)	: lengte van traject S volgens de oorspronkelijke nummering [hm]
END(I,J)	: nummer van het traject van het leidingennet van winput I dat direct grenst aan vraagpunt J [-]
G(S,0)	: aantal vraagpunten waarbij traject S deel uitmaakt van het leidingtracé naar deze vraagpunten [-]
G(S,JH)	: vraagpunten waarbij traject S deel uitmaakt van het leidingtracé naar deze vraagpunten [-]
I	: index van de winputten [-]
J	: index van de vraagpunten [-]
JD	: index van de dummievariabelen [-]
JH,JH1,JH2	: hulpindices [-]
K	: index van de spline functie intervallen [-]
KU	: aantal spline functie intervallen [-]
M	: aantal vraagpunten [-]
N	: aantal winputten [-]
ND	: totaal aantal dummievariabelen [-]
S	: index van de leidingtrajecten [-]
SH,SH1	: hulpindices [-]
SI(I)	: aantal trajecten van het leidingennet van winput I [-]
SIMAX	: maximum aantal trajecten dat in een leidingennet voorkomt [-]
SU	: aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
SIVAR1	: aantal leidingtrajecten (variabel), gebruikt bij het samenvoegen van trajecten, die naar dezelfde vraagpunt(en) leiden [-]
SIVAR2	: idem [-]

BIJLAGE X

Listing van het hoofdprogramma MAIN

```
      PROGRAM MAIN
C
C     Declareren van de parameters
C
      INTEGER CO,KU,M,N,NCP,ND,NTP,NVP,SU,TU
C
C     Toevoegen van het parameterbestand
C
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),AM(NCP,NVP),B(N),COEF(N,SU,M),COMB(ND,0:CO),
     #        END(N,M),I2,IU2(N),J2,JU2(M),NC,NV,QA,QBU(N,SU),SI(N),
     #        XB(NCP),X1(ND)
      REAL AB,AU(N),BE(TU),BI(NCP,NCP),BL(NCP),C(NVP),DUOPT,E(N,SU),
     #     F(N,M),G(N,M),GAM(TU,ND),H(N,KU),KQ(N,KU),PRIOPT,PSI(N,M),
     #     Q(N,KU),RES,RQ0(N),TCG,U0(M),X(NTP),XR(NVP),Z(N),ZR,ZZ
C
C     Declareren van de indices
C
      INTEGER I,J,T
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*62 AREA
C
C     Communiceren met de subroutines ANSOL, PRISOL, RESIM, TRANS
C     en ZEON
C
      COMMON/BLOCK1/ SI
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK3/ Q,QA,QBU
      COMMON/BLOCK4/ COMB,END
      COMMON/BLOCK5/ X1,TCG
      COMMON/BLOCK6/ PRIOPT,PSI,RES,X
      COMMON/BLOCK7/ AM,BI,BL,C,NC,NV,XB,XR,ZR
      COMMON/BLOCK8/ AB,I2,IU2,J2,JU2,U0,Z
      COMMOM/BLOCK9/ BE,GAM,DUOPT,ZZ
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(2,5) DATE
      READ(2,15) AREA
      READ(3,5) DATE
      READ(3,15) AREA
      WRITE(4,25) DATE,AREA
C
C     Uitvoeren van de gebruikte parameters
C
```

```
      WRITE(4,35)
      WRITE(4,45) CO,KU,M,N,SU,TU,NCP,ND,NTP,NVP
C
C     Aanroepen van de subroutines RECALCO en ANSOL
C
      CALL RECALCO
      CALL ANSOL
C
C     Uitvoeren van het in ANSOL berekende minimum TCG
C
      WRITE(4,55)
      WRITE(4,65) TCG
      T=0
      ZZ=0.0
   10 CONTINUE
C
C     De teller die het aantal iteraties bijhoudt met één verhogen
C
      T=T+1
C
C     Controleren of de teller het vooraf vastgestelde maximum aantal
C     iteraties overschrijdt
C
      IF(T.GT.TU) THEN
        WRITE(4,75)
        STOP
      ENDIF
C
C     Het nummer van de iteratie uitvoeren
C
      WRITE(4,85) T
C
C     Aanroepen van de subroutine PRISOL
C
      CALL PRISOL(*20,*40)
C
C     Uitvoeren van de in de subroutine PRISOL berekende minima
C
   20 CONTINUE
      WRITE(4,95) ZR
      WRITE(4,105) RES
      WRITE(4,115) PRIOPT
C
C     Testen of de optimale oplossing bereikt is en zo ja,
C     de berekende waarden uitvoeren en het programma beeindigen
C
      IF(ABS(PRIOPT-ZZ)/PRIOPT.LT.0.001)THEN
        WRITE(4,125) T
        DO 30 I=1,N
              WRITE(4,135) I, (X(J),J=(I-1)*M+1,I*M)
              WRITE(4,145) I, (X(J),J=N*M+(I-1)*M+1,N*M+I*M)
              WRITE(4,155) I, (X(J),J=2*N*M+(I-1)*KU+1,2*N*M+I*KU)
   30   CONTINUE
        WRITE(4,165) (X(J),J=2*N*M*+N*KU+1,2*N*M+N*KU+M)
        WRITE(4,175) (X(J),J=NTP-N+1,NTP)
        WRITE(4,185) PRIOPT
```

```
        STOP
      ENDIF
C
C     Aanroepen van de subroutine TRANS
C
      CALL TRANS(T)
C
C     Uitvoeren van het in de subroutine TRANS berekende maximum
C
      WRITE(4,195) DUOPT
C
C     Aanroepen van de subroutine ZEON
C
      CALL ZEON(T)
C
C     Uitvoeren van het in de subroutine ZEON berekende minimum
C
      WRITE(4,205) ZZ
C
C     Itereren
C
      GOTO 10
   40 CONTINUE
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A62)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,Gebruikte parameters:')
   45 FORMAT(/X,'CO',X,':',X,I2,5X,'KU:',X,I2,5X,'M',2X,':',
     #       X,I2,5X,'N',2X,':',X,I2,5X,'SU:',X,I2,5X,'TU:',
     #       X,I2//X,'NCP:',X,I2,5X,'ND:',X,I2,5X,'NTP:',X,
     #       I2,5X,'NVP:',X,I2)
   55 FORMAT(//X,'Berekende variabelen:')
   65 FORMAT(/X,'TCG',3X,':',X,F10.1)
   75 FORMAT(//X,'Het hoofdprogramma MAIN is gestopt omdat het ',
     #         'maximum aantal'/X,'iteraties bereikt is')
   85 FORMAT(//X,'Iteratie',X,I2':')
   95 FORMAT(/X,'ZR',4X,':',X,F10.1)
  105 FORMAT(/X,'RES',3X,':',X,F10.1)
  115 FORMAT(/X,'PRIOPT:',X,F10.1)
  125 FORMAT(//X,'Het in iteratie',X,I2,X,'bereikte optimum:')
  135 FORMAT(/X,'QW(',I1,',J):',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  145 FORMAT(/X,'QB(',I1,',J):',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  155 FORMAT(/X,'QX(',I1,',K):',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  165 FORMAT(/X,'YS0(J)',2X,':',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  175 FORMAT(/X,'YS1(I)',2X,':',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  185 FORMAT(/X,'TCG',3X,':',X,F10.1)
  195 FORMAT(/X,'DUOPT',X,':',X,F10.1)
  205 FORMAT(/X,'ZZ',4X,':',X,F10.1)
      END
```

Betekenis van de in het programma MAIN gebruikte variabelen

AREA : identificatie van het proefgebied [-]
CO : maximum aantal afhankelijke dummievariabelen [-]
DATE : datum van de programma-run [-]
DUOPT : maximum van het duale probleem, zoals berekend in de subroutine TRANS [f]
I : index van de winputten [-]
J : index van de dummievariabelen [-]
KU : aantal spline functie intervallen [-]
M : aantal vraagpunten [-]
N : aantal winputten [-]
NCP : aantal beperkingen van het volledige primale probleem [-]
ND : totaal aantal dummievariabelen [-]
NTP : totaal aantal variabelen van het volledige primale probleem (inclusief de spelingsvariabelen) [-]
NVP : aantal variabelen van het volledige primale probleem [-]
PRIOPT: minimum van het primale probleem, zoals berekend in de subroutine PRISOL [f]
RES : bijdrage van de dummievariabelen aan PRIOPT [f]
SU : aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
T : index van de iteratiestappen van het oplossingsalgorithme [-]
TCG : minimum van het primale probeem, zoals berekend in de subroutine ANSOL [f]
TU : maximum aantal iteratiestappen van het oplossingsalgorithme [-]
X(J) : continue variabelen [m^3]
ZR : bijdrage van de continue variabelen aan PRIOPT [f]
ZZ : minimum van het primale probleem, zoals berekend in de subroutine ZEON [f]

De betekenis van de overige variabelen wordt verklaard in de subroutines, waarin deze variabelen daadwerkelijk gebruikt worden.

BIJLAGE XI

Listing van de subroutine RECALCO

```
      SUBROUTINE RECALCO
C
C     Declareren van de parameters
C
      INTEGER CO,KU,M,N,ND,SU
C
C     Toevoegen van het parameterbestand
C
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),B(N),COEF(N,SU,M),COMB(ND,0:CO),END(N,M),QA,
     #        QBU(N,SU),SI(N)
      REAL AU(N),E(N,SU),F(N,M),G(N,M),H(N,KU),KQ(N,KU),Q(N,KU),RQ0(N)
C
C     Declareren van de lokale variabelen
C
      INTEGER ALFA(N),CA(N),CB(N),CW(N),DT(N,SU),DW(N,M)
      REAL D(N),DB(N,M),DELTA(N),PG(N),R(N,0:KU),RA(N,0:KU),RO,
     #     RQ(N,KU),T(N),V(N)
C
C     Declareren van de indices
C
      INTEGER I,J,JD,K,S
C
C     Communiceren met de subroutines ANSOL, PRISOL en TRANS
C
      COMMON/BLOCK1/ SI
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK3/ Q,QA,QBU
      COMMON/BLOCK4/ COMB,END
C
C     Definiëren van de waarde van RO
C
      DATA RO /0.01/
C
C     Inlezen van de gegevens van het proefgebied
C
      JD=0
      DO 30 I=1,N
            READ(2,5) SI(I)
            DO 10 J=1,M
                  READ(2,15) (COEF(I,S,J),S=1,SI(I))
   10       CONTINUE
            READ(2,25) (DT(I,S),S=1,SI(I))
            READ(2,5) (END(I,J),J=1,M)
            READ(2,5) (COMB(J,0),J=JD+1,JD+2+SI(I)+KU)
            DO 20 J=JD+2,JD+2+SI(I)+KU
                  READ(2,25) (COMB(J,JH),JH=1,COMB(J,0))
   20       CONTINUE
```

```
          JD=JD+2+SI(I)+KU
   30 CONTINUE
      DO 40 I=1,N
          READ(3,*) ALFA(I),CA(I),CB(I),CW(I),D(I),DELTA(I),PG(I),
     #              RQ0(I),T(I),V(I),(DW(I,J),J=1,M)
   40 CONTINUE
      READ (3,*) A
C
C     Berekenen van de coëfficiënten
C
      QA=0
      DO 50 J=1,M
          QA=QA+A(J)
   50 CONTINUE
      DO 130 I=1,N
          R(I,0)=(2*T(I)*V(I)+(6*T(I)*RQ0(I))**(1.0/3))**2
          RA(I,0)=4*T(I)*(6*T(I)*RQ0(I))**(-2.0/3)*(2*T(I)*V(I)+
     #              (6*T(I)*RQ0(I))**(1.0/3))
          AU(I)=PG(I)*(R(I,0)-RQ0(I)*RA(I,0))+CA(I)+CB(I)
          B(I)=CW(I)-CB(I)
          DO 60 S=1,SI(I)
              E(I,S)=ALFA(I)*DT(I,S)/10.0
   60     CONTINUE
          DO 80 J=1,M
              F(I,J)=PG(I)*RA(I,0)+2.6+0.35*DW(I,J)
              DB(I,J)=0.0
              DO 70 S=1,SI(I)
                  DB(I,J)=DB(I,J))+COEF(I,S,J)*DT(I,S)/10.0
   70         CONTINUE
              G(I,J)=PG(I)*RA(I,0)+0.66+DELTA(I)*DB(I,J)
   80     CONTINUE
          RQ(I,1)=4.0/3*D(I)**3*T(I)**2
          DO 90 K=1,KU
              R(I,K)=((RQ(I,K)/D(I)-T(I)**2*D(I)**2/3)**0.5+T(I)*
     #                D(I)+2*T(I)*V(I))**2
              RA(I,K)=(1+(T(I)*D(I)+2*T(I)*V(I))/(RQ(I,K)/D(I)-
     #                T(I)**2*D(I)**2/3)**0.5)/D(I)
              H(I,K)=PG(I)*(RA(I,K)-RA(I,K-1))
              IF(K.EQ.1)THEN
                KQ(I,1)=(R(I,1)-R(I,0)+RQ0(I)*RA(I,0)-RQ(I,1)*
     #                  RA(I,1))/(RA(I,0)-RA(I,1))
              ENDIF
              Q(I,K)=QA-KQ(I,K)
              IF(K.EQ.KU)GOTO 100
              KQ(I,K+1)=T(I)**2*D(I)**3/3+D(I)*(((R(I,K)*(D(I)*
     #                  RA(I,K)-1)*RO)**0.5+(1+RO)*(T(I)*D(I)+2*
     #                  T(I)*V(I)))/(D(I)*RA(I,K)-1-RO))**2
              RQ(I,K+1)=T(I)**2*D(I)**3/3+D(I)*(KQ(I,K+1)/D(I)-
     #                  T(I)**2*D(I)**2/3)**2*(RQ(I,K)/D(I)-
     #                  T(I)**2*D(I)**2/3)
   90     CONTINUE
  100     CONTINUE
          DO 120 S=1,SI(I)
              QBU(I,S)=0
              DO 110 J=1,M
                  QBU(I,S)=QBU(I,S)+COEF(I,S,J)*A(J)
```

```
  110                 CONTINUE
  120          CONTINUE
  130 CONTINUE
      RETURN
C
C     Format statements
C
    5 FORMAT(X,25I3)
   15 FORMAT(X,75I1)
   25 FORMAT(X,15I4)
      END
```

Betekenis van de in de subroutine RECALCO gebruikte variabelen

A(J)	: vraag naar zand van vraagpunt J [m^3]
AU(I)	: parameter voor de vaste kosten van het in produktie nemen van winput I [f]
ALFA(I)	: kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand van winput I [f/km]
B(I)	: parameter voor de vaste kosten van het transport per vrachtwagen vanuit winput I [f]
CA(I)	: vaste kosten van het in produktie nemen van winput I [f]
CB(I)	: vaste winningskosten van winput I bij transport per pijpleiding [f]
CO	: maximum aantal afhankelijke dummievariabelen [-]
COEF(I,S,J)	: 0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
COMB(J,0)	: aantal dummievariabelen dat afhankelijk is van dummievariabele X1(J) [-]
COMB(J,JH)	: combinatiematrix die aangeeft welke dummievariabelen afhankelijk zijn van variabele X1(J) [-]
CW(I)	: vaste winningskosten van winput I bij transport per vrachtwagen [f]
D(I)	: dikte van de laag winbaar zand van winput I [m]
DB(I,J)	: lengte van het totale leidingtracé dat winput I met vraagpunt J verbindt [km]
DELTA(I)	: kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand van winput I [f/km]
DT(I,S)	: lengte van traject S van het leidingennet van winput I [hm]
DW(I,J)	: gewogen transportafstand van winput I naar vraagpunt J bij transport per vrachtwagen [km]
E(I,S)	: parameter voor de vaste kosten van traject S van het leidingennet van winput I [f]
END(I,J)	: nummer van het traject van het leidingennet van winput I dat direct grenst aan vraagpunt J [-]
F(I,J)	: coëfficiënt voor de variabele kosten van het transport per vrachtwagen van winput I naar vraagpunt J [f/m^3]

G(I,J)	:	coëfficiënt voor de variabele kosten van het transport per pijpleiding van winput I naar vraagpunt J [f/m^3]
H(I,K)	:	coëfficiënt voor de reductie in variabele kosten wanneer de produktie van winput I een hoeveelheid KQ(I,K) overschrijdt [f/m^3]
I	:	index van de winputten [-]
J	:	index van de vraagpunten [-]
JD	:	index voor de nummering van de continue variabelen en de dummievariabelen [-]
K	:	index van de spline functie intervallen [-]
KQ(I,K)	:	zandproduktie van winput I waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KU	:	aantal spline functie intervallen [-]
M	:	aantal vraagpunten [-]
N	:	aantal winputten [-]
ND	:	totaal aantal dummievariabelen [-]
PG(I)	:	grondprijs van winput I [f/m^2]
Q(I,K)	:	QA-KQ(I,K) [m^3]
QA	:	totale vraag naar zand [m^3]
QBU(I,S)	:	maximale hoeveelheid zand die vanuit winput I getransporteerd wordt door leidingtraject S [m^3]
R(I,K)	:	terreinoppervlakte als functie van de zandproduktie RQ(I,K) van winput I [m^2]
RA(I,K)	:	afgeleide van de functie R(I,K) [m^{-1}]
RO	:	relatieve afwijking $\frac{RS(q)-R(q)}{R(q)} < RO$ [-]
RQ(I,K)	:	zandproduktie van winput I die correspondeert met het begin van het K-de spline functie interval [m^3]
RQ0(I)	:	minimaal te produceren hoeveelheid zand ingeval winput I in produktie genomen wordt [m^3]
S	:	index van de leidingtrajecten [-]
SI(I)	:	aantal trajecten van het leidingennet van winput I [-]
SU	:	aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
T(I)	:	talud van de winput I [-]
V(I)	:	dikte van de niet uit zand bestaande bovenlaag van winput I [m]

BIJLAGE XII

Listing van de subroutine ANSOL

```
      SUBROUTINE ANSOL
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,ND,SU
C
C     Toevoegen van het parameterbestand
C
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),B(N),COEF(N,SU,M),QA,QBU(N,SU),SI(N),X1(ND)
      REAL AU(N),E(N,SU),F(N,M),G(N,M),H(N,KU),KQ(N,KU),Q(N,KU),
     #     RQ0(N),TCG
C
C     Declareren van de lokale variabelen
C
      INTEGER II(N),NU(N,M)
      REAL DH1,DH2,DH3,GAM,MIN,PSI1,PSI2,QB(N,M),QHELP,QW(N,M),
     #     QX(N,KU),SUM1,SUM2,SUM3,ZETA(N,M)
C
C     Declareren van de indices
C
      INTEGER I,J,JD,K,S
C
C     Communiceren met het hoofdprogramma MAIN en de subroutines
C     RECALCO, PRISOL en TRANS
C
      COMMON/BLOCK1/ SI
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK3/ Q,QA,QBU
      COMMON/BLOCK5/ X1,TCG
C
C     Berekenen van NU(I,J), ZETA(I,J) en II(J)
C
      DO 40 I=1,N
            SUM1=0.0
            DO 10 K=1,KU
                  SUM1=SUM1+H(I,K)*Q(I,K)
   10       CONTINUE
            DO 30 J=1,M
                  SUM2=0.0
                  DO 20 S=1,SI(I)
                        SUM2=SUM2+COEF(I,S,J)*E(I,S)/QBU(I,S)
   20             CONTINUE
                  PSI1=F(I,J)+(AU(I)+B(I)+SUM1)/QA
                  PSI2=G(I,J)+(AU(I)+SUM1)/QA+SUM2
                  NU(I,J)=1
                  IF(PSI1.LT.PSI2)NU(I,J)=0
                  ZETA(I,J)=(1-NU(I,J))*PSI1+NU(I,J)*PSI2
```

```
   30        CONTINUE
   40        CONTINUE
      DO 60 J=1,M
            MIN=ZETA(1,J)
            II(J)=1
            DO 50 I=2,N
                  IF(ZETA(I,J).GE.MIN)GOTO 50
                  MIN=ZETA(I,J)
                  II(J)=I
   50       CONTINUE
   60 CONTINUE
C
C     Berekenen van de dummievariabelen X1(J) en de minimale
C     kosten TCG
      TCG=0.0
      JD=0
      DO 150 I=1,N
             X1(JD+1)=0
             X1(JD+2)=0
             DO 70 S=1,SI(I)
                   X1(JD+2+S)=0
   70        CONTINUE
             SUM3=0.0
             DO 80 J=1,M
                   QHELP=0.0
                   IF(II(J).EQ.I)QHELP=A(J)
                   QB(I,J)=NU(I,J)*QHELP
                   QW(I,J)=(1-NU(I,J))*QHELP
                   TCG=TCG+ZETA(I,J)*QHELP
                   SUM3=SUM3+QHELP
   80        CONTINUE
             DH1=SUM3/QA
             IF(DH1.EQ.0.0)GOTO 140
             X1(JD+1)=1
             DH2=0.0
             DO 90 J=1,M
                   DH2=DH2+QW(I,J)/QA
   90        CONTINUE
             IF(DH2.GT.0.0)X1(JD+2)=1
             IF(DH2.EQ.DH1)GOTO 120
             DO 110 S=1,SI(I)
                    DH3=0.0
                    DO 100 J=1,M
                           DH3=DH3+COEF(I,S,J)*QB(I,J)/QBU(I,S)
  100               CONTINUE
                    IF(DH3.GT.0.0)X1(JD+2+S)=1
  110       CONTINUE
  120       CONTINUE
            GAM=0.0
            IF(SUM3.LT.RQ0(I))GAM=RQ0(I)-SUM3
            DO 130 K=1,KU
                   J=JD+2+SI(I)+K
                   X1(J)=1
                   QX(I,K)=SUM3+GAM-KQ(I,K)
                   IF(QX(I,K).LT.0.0)THEN
                     X1(J)=0
```

```
                    QX(I,K)=0.0
                  ENDIF
130         CONTINUE
140         JD=JD+2+SI(I)+KU
150 CONTINUE
    RETURN
    END
```

Betekenis van de in de subroutine ANSOL gebruikte variabelen

A(J)	: vraag naar zand van vraagpunt J [m^3]
AU(I)	: parameter voor de vaste kosten van het in produktie nemen van winput I [f]
B(I)	: parameter voor de vaste kosten van het transport per vrachtwagen vanuit winput I [f]
COEF(I,S,J)	: 0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
DH1,DH2,DH3	: hulpvariabelen [-]
E(I,S)	: parameter voor de vaste kosten van traject S van het leidingennet van winput I [f]
F(I,J)	: coëfficiënt voor de variabele kosten van het transport per vrachtwagen van winput I naar vraagpunt J [f/m^3]
G(I,J)	: coëfficiënt voor de variabele kosten van het transport per pijpleiding van winput I naar vraagpunt J [f/m^3]
GAM	: hulpvariabele [m^3]
H(I,K)	: coëfficiënt voor de reductie in variabele kosten wanneer de produktie van winput I een hoeveelheid KQ(I,K) overschrijdt [f/m^3]
I	: index van de winputten [-]
II(J)	: hulpvariabele [-]
J	: index van de vraagpunten [-]
JD	: index van de dummievariabelen [-]
K	: index van de spline functie intervallen [-]
KQ(I,K)	: zandproduktie van winput I waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KU	: aantal spline functie intervallen [-]
M	: aantal vraagpunten [-]
MIN	: hulpvariabele [f/m^3]
N	: aantal winputten [-]
ND	: totaal aantal dummievariabelen [-]
NU(I,J)	: hulpvariabele [-]
PSI1	: ,, [f/m^3]
PSI2	: ,, [f/m^3]
Q(I,K)	: QA - KQ(I,K) [m^3]
QA	: totale vraag naar zand [m^3]
QB(I,J)	: hoeveelheid zand die per pijpleiding van winput I naar vraagpunt J wordt getransporteerd [m^3]
QBU(I,S)	: maximale hoeveelheid zand die vanuit winput I getransporteerd wordt door leidingtraject S [m^3]
QHELP	: hulpvariabele [m^3]
QW(I,J)	: hoeveelheid zand die per vrachtwagen van winput I naar vraagpunt J wordt getransporteerd [m^3]

QX(I,K)	: hoeveelheid zand die meer geproduceerd wordt dan de hoeveelheid die correspondeert met de K-de knik in de lineaire spline functie benadering [m^3]
RQ0(I)	: minimaal te produceren hoeveelheid zand ingeval winput I in produktie genomen wordt [m^3]
S	: index van de leidingtrajecten [-]
SI(I)	: aantal trajecten van het leidingennet van winput I [-]
SU	: aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
SUM1	: hulpvariabele [f]
SUM2	: ,, [f]
SUM3	: ,, [m^3]
TCG	: minimale totale kosten [f]
X1(J)	: dummievariabele [-]
ZETA(I,J)	: hulpvariabele [f/m^3]

BIJLAGE XIII

Listing van de subroutine PRISOL

```
      SUBROUTINE PRISOL(*,*)
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,NCP,ND,NTP,NVP,SU
C
C     Toevoegen van het parameterbestand
C
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),AM(NCP,NVP),B(N),COEF(N,SU,M),NC,NV,SI(N),XB(NCP),
     #        X1(ND)
      REAL AU(N),BI(NCP,NCP),BL(NCP),C(NVP),E(N,SU),F(N,M),G(N,M),
     #     H(N,KU),KQ(N,KU),PRIOPT,PSI(N,M),RES,RQ0(N),X(NTP),XR(NVP),
     #     ZR
C
C     Declareren van de lokale variabelen
C
      INTEGER ETA(N,M),II(M),IND,IO(N),KS,NIO
      REAL HELP1,HELP2,LARGE,MIN,SUM,SUM1,SUM2
C
C     Declareren van de indices
C
      INTEGER I,IH,IH1,IH2,J,JH,JH1,JH2,JD,K,S
C
C     Communiceren met het hoofdprogramma MAIN en de subroutines
C     RECALCO, ANSOL, RESIM en TRANS
C
      COMMON/BLOCK1/ SI
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK5/ X1,TCG
      COMMON/BLOCK6/ PRIOPT,PSI,RES,X
      COMMON/BLOCK7/ AM,BI,BL,C,NC,NV,XB,XR,ZR
C
C     Bepalen van de right hand side BL(J)
C
      DO 10 J=1,M
            BL(J)=A(J)
   10 CONTINUE
      NIO=0
      LARGE=99.999
      JD=0
      DO 40 I=1,N
            IF(X1(JD+1).EQ.0)GOTO 30
            NIO=NIO+1
            IO(NIO)=I
            BL(M+NIO)=RQ0(I)
            KS=0
            DO 20 K=1,KU
```

```
                KS=KS+X1(JD+2+SI(I)+K)
   20         CONTINUE
              IF(KS.NE.0)BL(M+NIO)=KQ(I,KS)
   30         JD=JD+2+SI(I)+KU
   40 CONTINUE
C
C     Berekenen van PSI(I,J) en C(J)
C
      DO 120 IH1=1,NIO
             I=IO(IH1)
             JD=0
             IF(I.EQ.1)GOTO 60
             DO 50 IH2=1,I-1
                   JD=JD+2+SI(IH2)+KU
   50        CONTINUE
   60        SUM=0.0
             DO 70 K=1,KU
                   SUM=SUM+X1(JD+2+SI(I)+K)*H(I,K)
   70        CONTINUE
             DO 110 J=1,M
                    ETA(I,J)=0
                    IF(X1(JD+2).EQ.0)GOTO 80
                    IF(F(I,J).LT.G(I,J))THEN
                      ETA(I,J)=1
                      PSI(I,J)=F(I,J)+SUM
                      GOTO 100
                    ENDIF
   80               DO 90 S=1,SI(I)
                          IF(COEF(I,S,J).EQ.0)GOTO 90
                          IF(X1(JD+2+S).EQ.1)GOTO 90
                          ETA(I,J)=X1(JD+2)
                          PSI(I,J)=ETA(I,J)*F(I,J)+SUM)+(1-ETA(I,J))*
     #                              LARGE
                          GOTO 100
   90               CONTINUE
                    ETA(I,J)=2
                    PSI(I,J)=G(I,J)+SUM
  100               C(IH1-1)*M+J)=PSI(I,J)
  110        CONTINUE
  120 CONTINUE
      DO 130 J=NIO*M+1,NIO*M+M+NIO
             C(J)=0.0
  130 CONTINUE
C
C     Berekenen van NC en NV
C
      NC=M+NIO
      NV=NIO*M+NC
C
C     Bepalen van AM(I,J)
C
      DO 150 I=1,NC
             DO 140 J=1,NV
                    AM(I,J)=0
  140        CONTINUE
  150 CONTINUE
```

```
      DO 170 I=1,M
             DO 160 JH=1,NIO
                    J=(JH-1)*M+I
                    AM(I,J)=1
160          CONTINUE
170   CONTINUE
      DO 190 I=M+1,NC
             DO 180 JH=1,M
                    J=(I-M-1)*M+JH
                    AM(I,J)=1
180          CONTINUE
190   CONTINUE
      DO 200 I=1,NC
             J=NIO*M+I
             AM(I,J)=-1
200   CONTINUE
C
C     Bepalen van de startbasis (XB(J))
C
      DO 210 J=1,NV
             X(J)=0.0
210   CONTINUE
      DO 230 J=1,M
             MIN=LARGE
             DO 220 IH=1,NIO
                    I=IO(IH)
                    IF(PSI(I,J).GE.MIN)GOTO 220
                    MIN=PSI(I,J)
                    II(J)=IH
220          CONTINUE
230   CONTINUE
      DO 240 J=1,M-1
             XB(J)=(II(J)-1)*M+J
             JH=NIO*M+M+II(J)
             X(JH)=X(JH)+BL(J)
240   CONTINUE
      DO 250 J=M+1,NC-1
             JH=NIO*M+J
             IF(X(JH).GT.BL(J))THEN
               XB(J)=JH
             ELSE
               XB(J)=(J-M)*M
               X(NIO*M+M)=X(NIO*M+M)+BL(J)
             ENDIF
250   CONTINUE
      SUM1=BL(M)-X(NIO*M+M)
      SUM2=BL(NC)-X(NV)
      IND=0
      XB(M)=NIO*M
      XB(NC)=NIO*M+M
      IF(SUM2.LE.SUM1)THEN
        IND=1
        XB(NC)=NV
      ENDIF
      IF(IND*SUM1+(1-IND)*SUM2).LE.0.0)THEN
        IND=2
```

```
        XB(M)=NIO*M+M
        XB(NC)=NV
      ENDIF
C
C     Bepalen van de inverse matrix BI(I,J)
C
      DO 270 J=1,NC
             DO 260 I=1,NC
                    BI(I,J)=AM(I,XB(J))
 260         CONTINUE
 270  CONTINUE
      DO 290 I=1,NIO-1
             DO 280 J=1,M-1
                    HELP1=BI(M+I,J)
                    BI(M+I,J)=-BI(M+I,M+I)*HELP1
                    BI(M,J)=BI(M,J)+HELP1*BI(M,M+I)
 280         CONTINUE
 290  CONTINUE
      IF(IND.EQ.0)THEN
        DO 300 J=1,M-1
               HELP1=BI(M,J)
               HELP2=BI(NC,J)
               BI(M,J)=-HELP2
               BI(NC,J)=-HELP1-HELP2
 300    CONTINUE
        DO 310 I=M+1,NC-1
               HELP1=BI(M,I)
               BI(M,I)=0.0
               BI(NC,I)=HELP1
 310    CONTINUE
        BI(M,M)=0.0
        BI(M,NC)=1.0
        BI(NC,M)=-1.0
        BI(NC,NC)=1.0
      ENDIF
      IF(IND.EQ.1)THEN
        DO 320 J=1,M-1
               HELP1=BI(M,J)
               BI(NC,J)=BI(NC,J)+HELP1
 320    CONTINUE
        DO 330 I=M+1,NC-1
               BI(M,I)=-BI(M,I)
               BI(NC,I)=BI(M,I)
 330    CONTINUE
        BI(M,M)=1.0
        BI(M,NC)=0.0
        BI(NC,M)=1.0
        BI(NC,NC)=-1.0
      ENDIF
      IF(IND.EQ.2)THEN
        DO 340 J=1,M-1
               BI(M,J)=-BI(M,J)
 340    CONTINUE
        BI(M,M)=-1.0
        BI(M,NC)=0.0
        BI(NC,M)=0.0
```

```
        BI(NC,NC)=-1.0
      ENDIF
C
C     Aanroepen van de subroutine RESIM
C
      CALL RESIM(*350,*360)
  350 RETURN 2
  360 CONTINUE
C
C     Berekenen van het minimum PRIOPT en de continue variabelen X(J)
C
      RES=0.0
      JD=0
      DO 390 I=1,N
             RES=RES+AU(I)*X1(JD+1)+B(I)*X1(JD+2)
             DO 370 S=1,SI(I)
                    RES=RES+E(I,S)*X1(JD+2+S)
  370        CONTINUE
             DO 380 K=1,KU
                    RES=RES-X1(JD+2+SI(I)+K)*H(I,K)*KQ(I,K)
  380        CONTINUE
             JD=JD+2+SI(I)+KU
  390 CONTINUE
      PRIOPT=ZR+RES
      DO 400 J=1,NV
             X(J)=XR(J)
  400 CONTINUE
      DO 410 J=NV+1,NTP
             X(J)=0.0
  410 CONTINUE
      DO 420 J=1,M
             JH1=2*N*M+N*KU+J
             JH2=NIO*M+J
             X(JH1)=X(JH2)
             X(JH2)=0.0
  420 CONTINUE
      DO 430 I=1,NIO
             IH1=2*N*M+N*KU+M+IO(I)
             IH2=NIO*M+M+I
             X(IH1)=X(IH2)
             X(IH2)=0.0
  430 CONTINUE
      DO 450 IH=1,NIO
             I=IO(IH)
             DO 440 J=1,M
                    JH1=N*M+(I-1)*M+J
                    JH2=(IH-1)*M+J
                    X(JH1)=X(JH2)
                    X(JH2)=0.0
  440        CONTINUE
  450 CONTINUE
      DO 500 IH1=1,NIO
             I=IO(IH1)
             SUM=0.0
             DO 460 J=1,M
                    JH1=N*M+(I-1)*M+J
```

```
                IF(X(JH1).EQ.0.0)GOTO 460
                SUM=SUM+X(JH1)
                IF(ETA(I,J).EQ.2)GOTO 460
                JH2=JH1-N*M
                X(JH2)=X(JH1)
                X(JH1)=0.0
460         CONTINUE
            JD=0
            IF(I.EQ.1)GOTO 480
            DO 470 IH2=1,I-1
                JD=JD+2+SI(IH2)+KU
470         CONTINUE
480         DO 490 K=1,KU
                IF(X1(JD+2+SI(I)+K).EQ.0)GOTO 500
                J=2*N*M+(I-1)*KU+K
                X(J)=SUM-KQ(I,K)
490         CONTINUE
500 CONTINUE
    RETURN 1
    END
```

Betekenis van de in de subroutine PRISOL gebruikte variabelen

A(J)	: vraag naar zand van vraagpunt J [m^3]
AM(I,J)	: coëfficiëntenmatrix van de beperkingen [-]
AU(I)	: parameter voor de vaste kosten van het in produktie nemen van winput I [f]
B(I)	: parameter voor de vaste kosten van het transport per vrachtwagen vanuit winput I [f]
BI(I,J)	: inverse van de coëfficiëntenmatrix van de basisvariabelen [-]
BL(I)	: right hand side van de beperkingen [m^3]
C(J)	: coëfficiënt van de variabelen in de doelstellingsfunctie [f/m^3]
COEF(I,S,J)	: 0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
E(I,S)	: parameter voor de vaste kosten van traject S van het leidingennet van winput I [f]
ETA(I,J)	: hulpvariabele [-]
F(I,J)	: coëfficiënt voor de variabele kosten van het transport per vrachtwagen van winput I naar vraagpunt J [f/m^3]
G(I,J)	: coëfficiënt voor de variabele kosten van het transport per pijpleiding van winput I naar vraagpunt J [f/m^3]
H(I,K)	: coëfficiënt voor de reductie in variabele kosten wanneer de produktie van winput I een hoeveelheid KQ(I,K) overschrijdt [f/m3]
HELP1	: hulpvariabele [-]
HELP2	: ,, [-]
I	: index van de winputten [-]
IH,IH1,IH2	: hulpindices [-]
II(J)	: hulpvariabele [-]
IND	: ,, [-]

IO(I)	:	,, [-]
J	:	index van de vraagpunten [-]
JH,JH1,JH2	:	hulpindices [-]
K	:	index van de spline functie intervallen [-]
KQ(I,K)	:	zandproduktie van winput I waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KS	:	hulpvariabele [-]
KU	:	aantal spline functie intervallen [-]
LARGE	:	,, [f/m^3]
M	:	aantal vraagpunten [-]
MIN	:	hulpvariabele [f/m^3]
N	:	aantal winputten [-]
NC	:	aantal beperkingen van het gereduceerde primale probleem [-]
NCP	:	aantal beperkingen van het volledige primale probleem [-]
NTP	:	totaal aantal variabelen van het volledige primale probleem (inclusief de spelingsvariabelen) [-]
NV	:	aantal continue variabelen van het gereduceerde primale probleem [-]
NVP	:	aantal continue variabelen van het volledige primale probleem [-]
PRIOPT	:	minimum van het primale probleem [f]
PSI(I,J)	:	hulpvariabele [f/m^3]
RES	:	bijdrage van de dummievariabelen aan PRIOPT [f]
RQ0(I)	:	minimaal te produceren hoeveelheid zand ingeval winput I in produktie genomen wordt [m^3]
S	:	index van de leidingtrajecten [-]
SI(I)	:	aantal trajecten van het leidingennet van winput I [-]
SU	:	aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
SUM	:	hulpvariabele [f/m^3]
SUM1	:	,, [m^3]
SUM2	:	,, [m^3]
X(J)	:	continue variabele [m^3]
X1(J)	:	dummievariabele [-]
XB(J)	:	variabele die aangeeft welke variabelen basisvariabele zijn [-]
XR(J)	:	continue variabele, berekend in de subroutine RESIM [m^3]
ZR	:	bijdrage van de continue variabelen aan PRIOPT [f]

BIJLAGE XIV

Listing van de subroutine RESIM

```
      SUBROUTINE RESIM(*,*)
C
C     Declareren van de parameters
C
      INTEGER NCP,NVP
C
C     Toevoegen van het parameterbestand
C
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van niet-lokale variabelen
C
      INTEGER A(NCP,NVP),NC,NV,XB(NCP)
      REAL B(NCP),BI(NCP,NCP),C(NVP),X(NVP),Z
C
C     Declareren van de lokale variabelen
C
      INTEGER IND
      REAL AA(NCP),BA,FRAC,HELP,HELP1,HELP2(NCP),PROD(NCP),SUM,Z1,Z2
C
C     Declareren van de indices
C
      INTEGER I,IT,IZ,J,K,R
C
C     Communiceren met het hoofdprogramma MAIN en de subroutine PRISOL
C
      COMMON/BLOCK7/ A,BI,B,C,NC,NV,XB,X,Z
C
C     Controleren of de startbasis toelaatbaar is (BI*B ≥ 0); zo niet,

C     terugkeren naar het hoofdprogramma MAIN en de uitvoering ervan
C     onderbreken
C
      DO 20 I=1,NC
            HELP=0.0
            DO 10 J=1,NC
                  HELP=HELP+BI(I,J)*B(J)
   10       CONTINUE
            IF(HELP.LT.0.0)THEN
              WRITE(4,5)
              RETURN 1
            ENDIF
   20 CONTINUE
C
C     Stap 1
C
      Z1=0.0
      IZ=0
      IT=0
   30 CONTINUE
      IT=IT+1
```

```
      K=0
      DO 50 J=1,NC
            PROD(J)=0.0
            DO 40 I=1,NC
                  PROD(J)=PROD(J)+C(XB(I))*BI(I,J)
   40       CONTINUE
   50 CONTINUE
      HELP=0.0001
      DO 70 J=1,NV
            SUM=0.0
            DO 60 I=1,NC
                  SUM=SUM+PROD(I)*A(I,J)
   60       CONTINUE
            SUM=SUM-C(J)
            IF(HELP.GE.SUM)GOTO 70
            HELP=SUM
            K=J
   70 CONTINUE
      IF(K.EQ.0)GOTO 200
C
C     Stap 2
C
      IND=1
      HELP=0.0
      R=0
      DO 100 I=1,NC
             AA(I)=0.0
             BA=0.0
             DO 80 J=1,NC
                   AA(I)=AA(I)+BI(I,J)*A(J,K)
                   BA=BA+BI(I,J)*B(J)
   80        CONTINUE
             IF(AA(I).LE.0.0)GOTO 100
             IF(IND.EQ.0)GOTO 90
             FRAC=BA/AA(I)
             IF(FRAC.LT.HELP)GOTO 100
             R=I
             IND=0
             GOTO 100
   90        CONTINUE
             HELP=BA/AA(I)
             IF(FRAC.LE.HELP)GOTO 100
             FRAC=HELP
             R=I
  100 CONTINUE
C
C     Controleren of een variabele gevonden kan worden die de basis
C      verlaat (R>0); zo niet, dan is één van de variabelen onbegrensd
en
C     kan geen eindige oplossing gevonden worden: terugkeren naar het
C     hoofdprogramma MAIN en de uitvoering ervan onderbreken
C
      IF(R.EQ.0)THEN
        WRITE(4,15)
        RETURN 1
      ENDIF
```

```
      XB(R)=K
C
C     Stap 3
C
      HELP1=AA(R)
      DO 110 I=1,NC
             AA(I)=-AA(I)/HELP1
  110 CONTINUE
      AA(R)=1.0/HELP1
      DO 120 J=1,NC
             HELP2(J)=BI(R,J)
  120 CONTINUE
      DO 140 I=1,NC
             IF(I.EQ.R)GOTO 140
             DO 130 J=1,NC
                    BI(I,J)=BI(I,J)+AA(I)*HELP2(J)
  130        CONTINUE
  140 CONTINUE
      I=R
      DO 150 J=1,NC
             BI(I,J)=AA(R)*HELP2(J)
  150 CONTINUE
C
C     Stap 4
C
      DO 160 J=1,NV
             X(J)=0.0
  160 CONTINUE
      DO 180 I=1,NC
             DO 170 J=1,NC
                    X(XB(I))=X(XB(I))+B(I,J)*B(J)
  170        CONTINUE
  180 CONTINUE
C
C     Berekenen van de huidige waarde van de doelstellingsfunctie (Z2)
C
      Z2=0.0
      DO 190 J=1,NV
             Z2=Z2+C(J)*X(J)
  190 CONTINUE
C
C     Het huidige minimum wordt vergeleken met het minimum van de
C     voorgaande iteratiestap van RESIM. Indien deze waarde tijdens
C     10 opeenvolgende iteratiestappen (IZ=10) gelijk blijft, wordt
C     aangenomen dat het optimum bereikt is.
C
      IF(Z2.EQ.Z1)THEN
        IZ=IZ+1
        IF(IZ.EQ.10)GOTO 200
        GOTO 30
      ENDIF
      Z1=Z2
      IZ=0
      GOTO 30
C
C     Berekenen van de variabelen X(J) en het minimum Z
```

```
C
      DO 210 J=1,NV
             X(J)=0.0
  210 CONTINUE
      DO 230 I=1,NC
             DO 220 J=1,NV
                    X(XB(I)=X(XB(I)+BI(I)*B(J)
  220        CONTINUE
  230 CONTINUE
      Z=0.0
      DO 240 J=1,NV
             Z=Z+C(J)*X(J)
  240 CONTINUE
C
C     Terugkeren naar de subroutine PRISOL
C
      RETURN 2
C
C     Format statements
C
    5 FORMAT(//X,'De uitvoering van het hoofdprogramma MAIN',
     #       'wordt onderbroken'/X,'omdat de startbasis van de',
     #        'subroutine RESIM niet toelaatbaar is')
   15 FORMAT(//X,'De uitvoering van het hoofdprogramma MAIN',
     #       'wordt onderbroken'/X,'omdat één van de variabelen',
     #       'onbegrensd is')
      END
```

Betekenis van de in de subroutine RESIM gebruikte variabelen

A(I,J)	:	coëfficiëntenmatrix van de beperkingen [-]
AA(I)	:	hulpvariabele [-]
B(J)	:	right hand side van de beperkingen [m^3]
BA	:	hulpvariabele [m^3]
BI(I,J)	:	inverse van de coëfficiëntenmatrix van de basisvariabelen [-]
C(J)	:	coëfficiënt van de variabelen in de doelstellingsfunctie [f/m^3]
FRAC	:	hulpvariabele [m^3]
HELP	:	,, [m^3]
HELP1	:	,, [-]
HELP2(J)	:	,, [-]
I	:	index van de beperkingen [-]
IND	:	hulpvariabele [-]
IT	:	hulpindex [-]
IZ	:	,, [-]
J	:	index van de variabelen [-]
K	:	index die aangeeft welke variabele basisvariabele wordt [-]
NC	:	aantal beperkingen van het gereduceerde primale probleem [-]
NV	:	aantal variabelen van het gereduceerde primale probleem [-]
PROD(J)	:	hulpvariabele [f/m^3]
R	:	index die aangeeft welke variabele de basis verlaat [-]

SUM	:	hulpvariabele [f/m^3]
X(J)	:	continue variabele [m^3]
XB(I)	:	variabele die aangeeft welke variabelen basisvariabele zijn [-]
Z	:	minimum [f]
Z1	:	hulpvariabele [f]
Z2	:	,, [f]

BIJLAGE XV

Listing van de subroutine TRANS

```
      SUBROUTINE TRANS(T)
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,ND,NTP,SU,TU
C
C     Toevoegen van het parameterbestand
C
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),B(N),COEF(N,SU,M),I2,IU2(N),J2,JU2(M),QA,QBU(N,SU),
     #        SI(N),X1(ND)
      REAL AB,AU(N),BE(TU),DUOPT,E(N,SU),F(N,M),G(N,M),GAM(TU,ND),
     #     H(N,KU),KQ(N,KU),PSI(N,M),Q(N,KU),RQ0(N),U0(M),X(NTP-M-N),
     #     YS0(M),YS1(N),Z(N)
C
C     Declareren van de lokale variabelen
C
      INTEGER GI(N),IND,J1(M),KS,NJ1,NJ2,NNV(M),NV(M,N),NVL
      REAL HELP1,HELP2,QX,SUM,U4(N,KU),U5(N,KU)
C
C     Declareren van de indices
C
      INTEGER I,IH1,IH2,J,JD,JH,JH1,JH2,K,S,T
C
C     Communiceren met het hoofdprogramma MAIN en de subroutines
C     RECALCO, ANSOL, PRISOL, CORRECT en ZEON
C
      COMMON/BLOCK1/ SI
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK3/ Q,QA,QBU
      COMMON/BLOCK5/ X1,TCG
      COMMON/BLOCK6/ PRIOPT,PSI,RES,X,YS0,YS1
      COMMON/BLOCK8/ AB,I2,IU2,J2,JU2,U0,Z
      COMMON/BLOCK9/ BE,GAM,DUOPT,ZZ
C
C     Bepalen van NNV(J) en NV(J,NNV(J))
C
      DO 20 J=1,M
            NNV(J)=0
            DO 10 I=1,N
                  IH1=(I-1)*M+J
                  IH2=N*M+IH1
                  IF((X(IH1)+X(IH2)).EQ.0.0)GOTO 10
                  NNV(J)=NNV(J)+1
                  NV(J,NNV(J))=I
   10       CONTINUE
   20 CONTINUE
```

```
C
C     Stap 1
C
      JD=0
      DO 50 I=1,N
            Z(I)=0.0
            GI(I)=1
            IF(X(JD+1).EQ.0)GOTO 40
            KS=0
            DO 30 K=1,KU
                  KS=KS+X1(JD+2+SI(I)+K)
   30       CONTINUE
            QX=X(2*N*M+(I-1)*KU+KS)
            J=JD+2+SI(I)+1
            IF(((1-X1(J))*YS1(I)+X1(J)*QX).GT.0.0)GOTO 40
            GI(I)=0
   40       CONTINUE
            JD=JD+2+SI(I)+KU
   50 CONTINUE
C
C     Stap 2
C
      NJ1=0
      DO 100 J=1,M
             U0(J)=0.0
             IF(YS0(J).GT.0.0)GOTO 80
             DO 60 JH=1,NNV(J)
                   I=NV(J,JH)
                   IF(GI(I).EQ.1)GOTO 70
   60        CONTINUE
             NJ1=NJ1+1
             J1(NJ1)=J
             GOTO 100
   70        CONTINUE
             U0(J)=PSI(I,J)-Z(I)
   80        CONTINUE
             DO 90 JH=1,NNV(J)
                   I=NV(J,JH)
                   Z(I)=PSI(I,J)-U0(J)
                   GI(I)=1
   90        CONTINUE
  100 CONTINUE
C
C     Stap 3 (inclusief herhaling stap 2)
C
      IF(NJ1.EQ.0)GOTO 250
      NVL=0
  110 CONTINUE
      NJ2=0
      IND=0
      DO 150 JH1=1,NJ1
             J=J1(JH1)
             DO 120 JH2=1,NNV(J)
                    I=NV(J,JH2)
                    IF(GI(I).EQ.1)GOTO 130
  120        CONTINUE
```

```
                NJ2=NJ2+1
                J1(NJ2)=J
                NVL=NVL+NNV(J)-1
                GOTO 150
  130           CONTINUE
                U0(J)=PSI(I,J)-Z(I)
                IND=1
                DO 140 JH2=1,NNV(J)
                       I=NV(J,JH2)
                       Z(I)=PSI(I,J)-U0(J)
                       G(I)=1
  140           CONTINUE
  150 CONTINUE
C
C     Stap 4
C
      IF(NJ2.EQ.0)GOTO 250
      NJ1=NJ2
      IF(IND.EQ.1)GOTO 110
      IF(NVL.EQ.0)THEN
        DO 160 JH=1,NJ1
               J=J1(JH)
               I=NV(J,1)
               U0(J)=PSI(I,J)
               GI(I)=1
  160   CONTINUE
        GOTO 250
      ENDIF
C
C     Stap 5 (inclusief herhaling stap 2)
C
  170 CONTINUE
      J=J1(NJ1)
      I=NV(J,1)
      U0(J)=PSI(I,J)
      NJ1=NJ1-1
      NVL=NVL-NNV(J)+1
      I2=0
      J2=1
      JU2(J2)=J
      DO 180 JH=1,NNV(J)
               I=NV(J,JH)
               Z(I)=PSI(I,J)-U0(J)
               I2=I2+1-GI(I)
               IF(GI(I).EQ.0)IU2(I2)=I
               GI(I)=1
               AB=Z(I)
               IF(AB.LT.0.0)CALL CORRECT
  180 CONTINUE
      IF(NJ1.EQ.0)GOTO 250
  190 CONTINUE
      NJ2=0
      IND=0
      DO 230 JH1=1,NJ1
               J=J1(JH1)
               DO 200 JH2=1,NNV(J)
```

```
                I=NV(J,JH2)
                IF(GI(I).EQ.1)GOTO 210
200         CONTINUE
            NJ2=NJ2+1
            J1(NJ2)=J
            GOTO 230
210         CONTINUE
            U0(J)=PSI(I,J)-Z(I)
            IND=1
            NVL=NVL-NNV(J)+1
            J2=J2+1
            JU2(J2)=J
            AB=-U0(J)
            IF(AB.GT.0.0)CALL CORRECT
            DO 220 JH2=1,NNV(J)
                   I=NV(J,JH2)
                   Z(I)=PSI(I,J)-U0(J)
                   I2=I2+1-GI(I)
                   IF(GI(I).EQ.0)IU2(I2)=I
                   GI(I)=1
                   AB=Z(I)
                   IF(AB.LT.0.0)CALL CORRECT
220         CONTINUE
230 CONTINUE
    IF(NJ2.EQ.0)GOTO 250
    NJ1=NJ2
    IF(IND.EQ.1)GOTO 190
    IF(NVL.GT.0)GOTO 170
    DO 240 JH=1,NJ2
           J=J1(JH)
           I=NV(J,1)
           U0(J)=PSI(I,J)-Z(I)
           GI(I)=1
240 CONTINUE
C
C   Stap 6
C
250 CONTINUE
    BE(T)=0.0
    DO 260 J=1,M
           BE(T)=BE(T)+U0(J)*A(J)
260 CONTINUE
    DUOPT=BE(T)
    JD=0
    DO 300 I=1,N
           KS=0
           DO 270 K=1,KU
                  KS=KS+X1(JD+2+SI(I)+K)
                  U4(I,K)=0.0
                  U5(I,K)=-H(I,K)
270        CONTINUE
           IF(KS.EQ.0)GOTO 290
           DO 280 K=1,KS
                  U4(I,K)=-H(I,K)
                  U5(I,K)=0.0
280        CONTINUE
```

```
          U4(I,KS)=U4(I,KS)+Z(I)
290       CONTINUE
          JD=JD+2+SI(I)+KU
300 CONTINUE
    JD=0
    DO 370 I=1,N
          HELP1=0.0
          JH=JD+1
          IF(X1(JH).EQ.0)THEN
            HELP1=U0(1)-F(I,1)
            IF(G(I,1).LT.F(I,1))HELP1=U0(1)-G(I,1)
            DO 310 J=2,M
                   HELP2=U0(J)-F(I,J)
                   IF(G(I,J).LT.F(I,J))HELP2=U0(J)-G(I,J)
                   IF(HELP2.GT.HELP1)HELP1=HELP2
310         CONTINUE
            IF(HELP1.LT.0.0)HELP1=0.0
          ENDIF
          GAM(T,JH)=AU(I)+RQ0(I)*(1-X1(JD+2+SI(I)+1))*Z(I)-HELP1*QA
          DUOPT=DUOPT+X1(JH)*GAM(T,JH)
          HELP1=0.0
          JH=JD+2
          SUM=0.0
          IF(X1(JD+2+SI(I)+1).EQ.1)THEN
            DO 320 K=1,KU
                   SUM=SUM+X1(JD+2+SI(I)+K)*H(I,K)
320         CONTINUE
          ENDIF
          IF(X1(JH).EQ.0)THEN
            HELP1=U0(1)-F(I,1)-SUM
            DO 330 J=2,M
                   HELP2=U0(J)-F(I,J)-SUM
                   IF(HELP2.GT.HELP1)HELP1=HELP2
330         CONTINUE
            IF(HELP1.LT.0.0)HELP1=0.0
          ENDIF
          GAM(T,JH)=B(I)-HELP1*QA
          DUOPT=DUOPT+X1(JH)*GAM(T,JH)
          DO 350 S=1,SI(I)
                 HELP1=0.0
                 JH=JD+2+S
                 IF(X1(JH).EQ.0)THEN
                   HELP1=COEF(I,S,1)*(U0(1)-G(I,1)-SUM)
                   DO 340 J=2,M
                          HELP2=COEF(I,S,J)*(U0(J)-G(I,J)-SUM)
                          IF(HELP2.GT.HELP1)HELP1=HELP2
340                CONTINUE
                   IF(HELP1.LT.0.0)HELP1=0.0
                 ENDIF
                 GAM(T,JH)=E(I,S)-HELP1*QBU(I,S)
                 DUOPT=DUOPT+X1(JH)*GAM(T,JH)
350       CONTINUE
          DO 360 K=1,KU
                 JH=JD+2+SI(I)+K
                 GAM(T,JH)=KQ(I,K)*U4(I,K)-Q(I,K)*U5(I,K)
                 DUOPT=DUOPT+X1(JH)*GAM(T,JH)
```

```
360         CONTINUE
            JD=JD+2+SI(I)+KU
370 CONTINUE
    RETURN
    END
```

Betekenis van de in de subroutine TRANS gebruikte variabelen

A(J)	: vraag naar zand van vraagpunt J [m^3]
AB	: waarde van de duale variabele [f/m^3]
AU(I)	: parameter voor de vaste kosten van het in produktie nemen van winput I [f]
B(I)	: parameter voor de vaste kosten van het transport per vrachtwagen vanuit winput I [f]
BE(T)	: constante van de beperking T in de subroutine ZEON [f]
COEF(I,S,J)	: 0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
DUOPT	: maximum van het duale probleem [f]
E(I,S)	: parameter voor de vaste kosten van traject S van het leidingennet van winput I [f]
F(I,J)	: coëfficiënt voor de variabele kosten van het transport per vrachtwagen van winput I naar vraagpunt J [f/m^3]
G(I,J)	: coëfficiënt voor de variabele kosten van het transport per pijpleiding van winput I naar vraagpunt J [f/m^3]
GAM(T,J)	: coëfficiënten van de dummievariabelen van de beperking T in de subroutine ZEON [f]
GI(I)	: 0-1 variabele die aangeeft of Z(I) gewaardeerd is [-]
H(I,J)	: coëfficiënt voor de reductie in variabele kosten wanneer de produktie van winput I een hoeveelheid KQ(I,K) overschrijdt [f/m^3]
HELP1	: hulpvariabele [f/m^3]
HELP2	: ,, [f/m^3]
I	: index van de winputten [-]
I2	: hulpindex [-]
IH1,IH2	: hulpindices [-]
IND	: 0-1 variabele die aangeeft of een U0(J) in de huidige iteratiestap van TRANS gewaardeerd is [-]
IU2(I)	: hulpvariabele [-]
J	: index van de vraagpunten [-]
J1(J)	: verzameling indices J van de variabelen U0(J) die nog niet gewaardeerd zijn [-]
J2	: hulpindex [-]
JH,JH1,JH2	: hulpindices [-]
JU2(J)	: hulpvariabele [-]
K	: index van de spline functie intervallen [-]
KQ(I,K)	: zandproduktie van winput I waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KS	: $\sum_{K=1}^{KU} D4(I,K)$

KU	:	aantal spline functie intervallen [-]
M	:	aantal vraagpunten [-]
N	:	aantal winputten [-]
ND	:	totaal aantal dummievariabelen
NJ1	:	aantal indices J1(J) [-]
NJ2	:	aantal indices J1(J) [-]
NNV(J)	:	aantal winputten waaruit zand vervoerd wordt naar vraagpunt J [-]
NTP	:	totaal aantal variabelen van het volledige primale probleem (inclusief de spelingsvariabelen) [-]
NV(J,I)	:	nummer van de winputten waaruit zand vervoerd wordt naar vraagpunt J [-]
NVL	:	aantal variabelen U0(J) dat nog niet gewaardeerd is minus het aantal vergelijkingen dat nog niet gebruikt is om een variabele U0(J) of Z(I) te waarderen [-]
PSI(I,J)	:	hulpvariabele [f/m^3]
Q(I,K)	:	QA - KQ(I,K) [m^3]
QA	:	totale vraag naar zand [m^3]
QBU(I,S)	:	maximale hoeveelheid zand die vanuit winput I getransporteerd wordt door leidingtraject S [m^3]
QX	:	hulpvariabele [m^3]
RQ0(I)	:	minimaal te produceren hoeveelheid zand ingeval winput I in produktie genomen wordt [m^3]
S	:	index van de leidingtrajecten [-]
SI(I)	:	aantal trajecten van het leidingennet van winput I [-]
SU	:	aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
T	:	index van de iteratiestappen van het oplossingsalgorithme [-]
TU	:	maximum aantal iteratiestappen van het oplossingsalgorithme [-]
SUM	:	hulpvariabele [f/m^3]
U0(J)	:	duale variabele [f/m^3]
U4(I,K)	:	duale spelingsvariabele [f/m^3]
U5(I,K)	:	duale spelingsvariabele [f/m^3]
X(J)	:	primale variabele [m^3]
X1(J)	:	dummievariabele [-]
YS0(J)	:	primale spelingsvariabele [m^3]
YS1(I)	:	primale spelingsvariabele [m^3]
Z(I)	:	duale variabele [f/m^3]

Listing van de subroutine CORRECT

```
      SUBROUTINE CORRECT
C
C     Declareren van de parameters
C
      INTEGER M,N
C
C     Toevoegen van het parameterbestand
C
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER I2,IU2(N),J2,JU2(M)
      REAL AB,U0(M),Z(N)
C
C     Declareren van de indices
C
      INTEGER I,IH,J,JH
C
C     Communiceren met de subroutine TRANS
C
      COMMON/BLOCK8/ AB,I2,IU2,J2,JU2,U0,Z
C
      DO 10 IH=1,I2
            I=IU2(IH)
            Z(I)=Z(I)-AB
   10 CONTINUE
      DO 20 JH=1,J2
            J=JU2(JH)
            U0(J)=U0(J)+AB
   20 CONTINUE
      RETURN
      END
```

Betekenis van de in de subroutine CORRECT gebruikte variabelen

AB	:	waarde van de duale variabele [f/m^3]
I	:	index van de winputten [-]
I2	:	hulpindex [-]
IU2(I)	:	hulpvariabele [-]
J	:	index van de vraagpunten [-]
J2	:	hulpindex [-]
JU2(J)	:	hulpvariabele [-]
M	:	aantal vraagpunten [-]
N	:	aantal winputten [-]
U0(J)	:	duale variabele [f/m^3]
Z(I)	:	duale variabele [f/m^3]

BIJLAGE XVII

Listing van de subroutine ZEON

```
      SUBROUTINE ZEON(TZ)
C
C     Declareren van de parameters
C
      INTEGER CO,KU,M,N,ND,TU
C
C     Toevoegen van het parameterbestand
C
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER COMB(ND,0:CO),END(N,M),SI(N),TZ,X1(ND)
      REAL BE(TU),GAM(TU,ND),Z
C
C     Declareren van de lokale variabelen
C
      INTEGER G(ND),GN,IN(ND),IND,J1,L1,MMAX,MU,MUT(ND,0:CO),NMUT,
     #        SUM,T1,XS(ND)
      REAL AH(0:1),AL(TU),AM(0:1),AS,FI(ND),HELP,MAX,MIN,PROD
C
C     Declareren van de indices
C
      INTEGER I,J,JD,JH,JH1,JH2,S,T
C
C     Communiceren met het hoofdprogramma MAIN en de subroutines
C     RECALCO, PRISOL en TRANS
C
      COMMON/BLOCK1/ SI
      COMMON/BLOCK4/ COMB,END
      COMMON/BLOCK5/ X1,TCG
      COMMON/BLOCK9/ BE,GAM,DUOPT,Z
C
C     Stap 0
C
      MU=0
      GN=0
      AS=BE(1)
      DO 10 T=1,TZ
            IF(BE(T).GT.AS)AS=BE(T)
   10 CONTINUE
      DO 30 J=1,ND
            FI(J)=0.0
            G(J)=1
            IND=0
            XS(J)=0
            X1(J)=0
            DO 20 T=1,TZ
                  IF(GAM(T,J).GT.0.0)GOTO 30
                  IF(GAM(T,J).LT.0.0)IND=1
   20       CONTINUE
```

```
            IF(IND.EQ.0)GOTO 30
            IF(COMB(J,0).EQ.0)THEN
              G(J)=0
              X1(J)=1
              GOTO 30
            ENDIF
            G(J)=2
            GN=GN+1
   30 CONTINUE
C
C     Stap 1
C
   40 CONTINUE
      IF(GN.EQ.0)GOTO 90
      NMUT=0
      DO 80 J=1,ND
            IF(G(J).LT.2)GOTO 80
            DO 60 T=1,TZ
                  HELP=GAM(T,J)
                  DO 50 JH2=1,COMB(J,0)
                        JH1=COMB(J,JH2)
                        HELP=HELP+(1-X1(JH1))*GAM(T,JH1)
   50             CONTINUE
                  IF(HELP.GT.0)GOTO 80
   60       CONTINUE
            GN=GN-1
            NMUT=NMUT+1
            G(J)=0
            X1(J)=1
            DO 70 JH2=1,COMB(J,0)
                  JH1=COMB(J,JH2)
                  IF(G(JH1).EQ.2)GN=GN-1
                  G(JH1)=0
                  X1(JH1)=1
   70       CONTINUE
   80 CONTINUE
      IF(NMUT.GT.0)GOTO 40
C
C     Stap 2
C
   90 CONTINUE
      IF(TZ.EQ.1)GOTO 330
      MMAX=0
      DO 100 J=1,ND
             IF(G(J).EQ.2)G(J)=1
             MMAX=MMAX+G(J)
  100 CONTINUE
  110 CONTINUE
      MU=MU+1
      AL(1)=BE(1)
      DO 120 J=1,ND
             AL(1)=AL(1)+X1(J)*GAM(1,J)
  120 CONTINUE
      MAX=AL(1)
      T1=1
      DO 140 T=2,TZ
```

```
            AL(T)=BE(T)
            DO 130 J=1,ND
                  AL(T)=AL(T)+X1(J)*GAM(T,J)
130         CONTINUE
            IF(AL(T).LT.MAX)GOTO 140
            MAX=AL(T)
            T1=T
140   CONTINUE
      IF(MAX.LT.AS)THEN
        AS=MAX
        DO 150 J=1,ND
              XS(J)=X1(J)
150     CONTINUE
      ENDIF
C
C     Stap 3
C
      IF(MU.GT.MMAX)GOTO 250
      MIN=0.0
      DO 170 J=1,ND
            PROD=G(J)*GAM(T1,J)
            IF(PROD.GE.0.0)GOTO 170
            IF(COMB(J,0).GT.0)THEN
              DO 160 JH2=1,COMB(J,0)
                    JH1=COMB(J,JH2)
                    PROD=PROD+G(JH1)*GAM(T1,JH1)
160           CONTINUE
            ENDIF
            IF(PROD.GE.MIN)GOTO 170
            MIN=PROD
            J1=J
            IN(MU)=J1
170   CONTINUE
      IF(MIN.EQ.0)GOTO 250
C
C     Stap 4
C
      G(J1)=0
      AH(0)=AL(1)
      DO 180 J=1,ND
            IF(GAM(1,J).LT.0.0)AH(0)=AH(0)+G(J)*GAM(1,J)
180   CONTINUE
      AH(1)=AH(0)+GAM(1,J1)
      IF(COMB(J1,0).GT.0)THEN
        DO 190 JH=1,COMB(J1,0)
              J=COMB(J1,JH)
              IF(G(J)*GAM(1,J).LT.0.0)GOTO 190
              AH(1)=AH(1)+G(J)*GAM(1,J)
190     CONTINUE
      ENDIF
      AM(0)=AH(0)
      AM(1)=AH(1)
      DO 220 T=2,TZ
            AH(0)=AL(T)
            DO 200 J=1,ND
                  IF(GAM(T,J).LT.0.0) AH(0)=AH(0)+G(J)*GAM(T,J)
```

```
200         CONTINUE
            AH(1)=AH(0)+GAM(T1,J)
            IF(COMB(J1,0).GT.0)THEN
              DO 210 JH=1,COMB(J1,0)
                     J=COMB(J1,JH)
                     IF(G(J)*GAM(T,J).LT.0.0)GOTO 210
                     AH(1)=AH(1)+G(J)*GAM(T,J)
210           CONTINUE
            ENDIF
            IF(AH(0).GT.AM(0))AM(0)=AH(0)
            IF(AH(1).GT.AM(1))AM(1)=AH(1)
220 CONTINUE
    L1=1
    IF(AM(0).LT.AM(1))L1=0
    X1(J1)=L1
    FI(J1)=AM(1-L1)
    IF(COMB(J1,0)*L1.GT.0)THEN
      DO 230 JH=1,COMB(J1,0)
             J=COMB(J1,JH)
             IF(G(J)+X1(J).GT.0)GOTO 230
             L1=0
             X1(J1)=0
             FI(J1)=AS+1
230   CONTINUE
    ENDIF
    IF(AM(L1).GT.AS)THEN
      G(J1)=1
      X1(J1)=0
      GOTO 250
    ENDIF
    MUT(MU,0)=0
    IF(COMB(J1,0)*L1.EQ.0)GOTO 110
    DO 240 JH2=1,COMB(J1,0)
            JH1=COMB(J1,JH2)
            IF(G(JH1).EQ.0)GOTO 240
            MUT(MU,0)=MUT(MU,0)+1
            J=MUT(MU,0)
            MUT(MU,J)=JH1
            X1(JH1)=1
            G(JH1)=0
240 CONTINUE
    GOTO 110
C
C     Stap 5
C
250 CONTINUE
    MU=MU-1
    IF(MU.EQ.0)GOTO 310
    J1=IN(MU)
    IF(FI(J1).GE.AS)THEN
      G(J1)=1
      X1(J1)=0
      IF(MUT(MU,0).EQ.0)GOTO 250
      DO 260 JH=1,MUT(MU,0)
             J=MUT(J,JH)
             G(J)=1
```

```
                X1(J)=0
                MUT(MU,JH)=0
260       CONTINUE
          MUT(MU,0)=0
          GOTO 250
        ENDIF
        X1(J1)=1-X1(J1)
        FI(J1)=AS+1.0
        IF(X1(J1).EQ.0)THEN
          IF(MUT(MU,0).EQ.0)GOTO 110
          DO 270 JH=1,MUT(MU,0)
                J=MUT(MU,JH)
                G(J)=1
                X1(J)=0
                MUT(MU,JH)=0
270       CONTINUE
          MUT(MU,0)=0
          GOTO 110
        ENDIF
        MUT(MU,0)=0
        IF(COMB(J1,0).EQ.0)GOTO 110
        DO 300 JH1=1,COMB(J1,0)
               J=COMB(J1,JH1)
               IF(G(J).EQ.1)GOTO 290
               IF(X1(J).EQ.1)GOTO 300
               FI(J1)=AS+1.0
               G(J1)=1
               X1(J1)=0
               IF(MUT(MU,0).EQ.0)GOTO 250
               DO 280 JH2=1,MUT(MU,0)
                      J=MUT(MU,JH2)
                      G(J)=1
                      X1(J)=0
                      MUT(MU,JH)=0
280            CONTINUE
               MUT(MU,0)=0
               GOTO 250
290            CONTINUE
               MUT(MU,0)=MUT(MU,0)+1
               JH2=MUT(MU,0)
               MUT(MU,JH2)=J
               G(J)=0
               X1(J)=1
300     CONTINUE
        GOTO 110
C
C       Stap 6
C
310     CONTINUE
        Z=AS
        DO 320 J=1,ND
               X1(J)=XS(J)
320     CONTINUE
        GOTO 350
```

```
C
C     Stap 7
C
  330 CONTINUE
      Z=BE(1)
      DO 340 J=1,ND
             Z=Z+X1(J)*GAM(1,J)
  340 CONTINUE
C
C     Stap 8
C
  350 CONTINUE
      IND=0
      SUM=0
      JD=0
      DO 360 I=1,N
             SUM=SUM+X1(JD+1)
             JD=JD+2+SI(I)+KU
  360 CONTINUE
      IF(SUM.EQ.0)THEN
        X1(1)=1
        X1(2)=1
        GOTO 430
      ENDIF
      JD=0
      DO 400 I=1,N
             IF(X1(JD+1).EQ.1)THEN
               IF(X1(JD+2).EQ.1)GOTO 380
               DO 370 J=1,M
                      S=END(I,J)
                      IF(X1(JD+2+S).EQ.1)GOTO 390
  370          CONTINUE
               X1(JD+2)=1
  380          CONTINUE
               IND=1
             ENDIF
  390        CONTINUE
             JD=JD+2+SI(I)+KU
  400 CONTINUE
      IF(IND.EQ.1)GOTO 430
      DO 420 J=1,M
             JD=0
             DO 410 I=1,N
                    IF(X1(JD+1).EQ.1)THEN
                      S=END(I,J)
                      IF(X1(JD+2+S).EQ.1)GOTO 420
                      X1(JD+2)=1
                      GOTO 420
                    ENDIF
                    JD=JD+2+SI(I)+KU
  410        CONTINUE
  420 CONTINUE
C
C     Stap 9
C
  430 CONTINUE
```

```
      AL(1)=BE(1)
      DO 440 J=1,ND
             AL(1)=AL(1)+X1(J)*GAM(1,J)
440   CONTINUE
      MAX=AL(1)
      IF(TZ.EQ.1)GOTO 470
      DO 460 T=2,TZ
             AL(T)=BE(T)
             DO 450 J=1,ND
                    AL(T)=AL(T)+X1(J)*GAM(T,J)
450          CONTINUE
             IF(AL(T).GT.MAX)MAX=AL(T)
460   CONTINUE
470   CONTINUE
      Z=MAX
      RETURN
      END
```

Betekenis van de in de subroutine ZEON gebruikte variabelen

AH(0:1) : hulpvariabele [f]
AL(T) : de waarde van beperking T [f]
AM(0) : $\max_T \{AH(0)\}$ [f]
AM(1) : $\max_T \{AH(1)\}$ [f]
AS : de minimumwaarde van Z die tot dan toe bereikt is [f]
BE(T) : constante van de beperking T [f]
CO : maximum aantal afhankelijke dummievariabelen X1(J) [-]
COMB(J,0) : aantal dummievariabelen dat afhankelijk is van dummievariabele X1(J) [-]
COMB(J,JH): combinatiematrix die aangeeft welke dummievariabelen afhankelijk zijn van dummievariabele X1(J) [-]
END(I,J) : nummer van het traject van het leidingennet van winput I dat direct grenst aan vraagpunt J [-]
FI(J) : minimumwaarde voor Z die bereikt kan worden met X1(J) = 1 - X1(J) voor G(J) = 0 [f]
G(J) : variabele die aangeeft welke dummievariabele X1(J) gewaardeerd is [-]
GAM(T,J) : coëfficiënten van de dummievariabelen in beperking T [f]
GN : aantal dummievariabelen waarvoor geldt GAM(T,J)≤0 met tenminste één GAM(T,J) < 0 [-]
HELP : hulpvariabele [-]
I : index van de winputten [-]
IN(MU) : variabele die aangeeft welke X1(J) in stap MU pivotvariabele was [-]
IND : hulpvariabele [-]
J : index van de dummievariabelen [-]
J1 : index die aangeeft welke variabel pivotvariabele is [-]
JD : index van de dummievariabelen [-]
JH,JH1,JH2: hulpindices [-]

KU	:	aantal spline functie intervallen [-]
L1	:	AM(0) < AM(1): L1 = 0, anders L1 = 1 [-]
M	:	aantal vraagpunten [-]
MAX	:	hulpvariabele [-]
MMAX	:	,, [-]
MIN	:	,, [-]
MU	:	nummer van de iteratiestap binnen de subroutine ZEON [-]
MUT(J,0)	:	aantal dummievariabelen dat afhankelijk is van X1(J) en in de betreffende iteratiestap gewaardeerd is [-]
MUT(J,JH)	:	nummer van de dummievariabele die afhankelijk is van X1(J) en in de betreffende iteratiestap gewaardeerd wordt [-]
NMUT	:	aantal combinaties dat in stap 1 gewaardeerd wordt [-]
N	:	aantal winputten [-]
ND	:	totaal aantal dummievariabelen [-]
PROD	:	hulpvariabele [-]
S	:	index van de leidingtrajecten [-]
SI(I)	:	aantal trajecten van het leidingennet van winput I [-]
SUM	:	hulpvariabele [-]
T	:	index van de iteratiestappen van het oplossingsalgorithme (≈ nummer van de vergelijkingen) [-]
T1	:	index voor de pivotrij [-]
TU	:	maximum aantal iteratiestappen van het volledige oplossingsalgorithme [-]
TZ	:	aantal beperkingen in ZEON [-]
X1(J)	:	dummievariabele [-]
XS(J)	:	dummievariabele behorende bij AS [-]
Z	:	waarde van de oplossing van ZEON [f]

BIJLAGE XVIII

Listing van het programma SPLINE

```
      PROGRAM SPLINE
C
C     Declareren van de parameter
C
      INTEGER KU
C
C     Definiëren van de parameter
C
      PARAMETER (KU=5)
C
C     Declareren van de variabelen
C
      REAL D,KQ(KU),RO,R(0:KU),RA(0:KU),RQ(KU),T,V
C
C     Declareren van de indices
C
      INTEGER K
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*52 EXAMPLE
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) EXAMPLE
      WRITE(2,25) DATE,EXAMPLE
C
C     Inlezen van de invoergegevens
C
      READ(1,*) D,RO,RQ0,T,V
C
C     Berekenen van de spline functie variabelen
C
      R(0)=(2*T*V+(6*T*RQ0)**(1.0/3))**2
      RA(0)=4*T*(6*T*RQ0)**(-2.0/3)*(2*T*V+(6*T*RQ0)**(1.0/3))
      RQ(1)=4.0/3*D**3*T**2
      DO 10 K=1,KU
            R(K)=((RQ(K)/D-T**2*D**2/3)**0.5+T*D+2*T*V)**2
            RA(K)=(1+(T*D+2*T*V)/(RQ(K)/D-T**2*D**2/3)**0.5)/D
            IF(K.EQ.1)THEN
              KQ(1)=(R(1)-R(0)+RQ0*RA(0)-RQ(1)*RA(1))/(RA(0)-RA(1))
            ENDIF
            IF(K.EQ.KU)GOTO 10
            KQ(K+1)=T**2*D**3/3+D*(((R(K)*(D*RA(K)-1)*RO)**0.5+(1+RO)*
                    (T*D+2*T*V))/(D*RA(K)-1-RO))**2
     #              T**2*D**2/3)**0.5)**2
            RQ(K+1)=T**2*D**3/3+D*(KQ(K+1)/D-T**2*D**2/3)**2/(RQ(K)/D-

     #              T**2*D**2/3)
```

```
   10 CONTINUE
C
C     Uitvoeren van de gebruikte parameters, de ingelezen gegevens en
C     de berekende variabelen
C
      WRITE(2,35)
      WRITE(2,45) KU
      WRITE(2,55)
      WRITE(2,65) D,RO,RQ0,T,V
      WRITE(2,75)
      WRITE(2,85) RQ0,R(0)
      DO 20 K=1,KU
            WRITE(2,95)K,KQ(K),K,RQ(K),K,R(K)
   20 CONTINUE
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A52)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,'Gebruikte parameter:')
   45 FORMAT(/X,'KU:',X,I1)
   55 FORMAT(//X,'Ingelezen gegevens:')
   65 FORMAT(/X,'D:',X,F4.1,3X,'RO:',X,F4.2,3X,'RQ0:',X,F7.1,3X,'T:',
     #       X,F3.1,3X,'V:',X,F3.1)
   75 FORMAT(//X,'Berekende variabelen:')
   85 FORMAT(/20X,'RQ0',2X,':',X,F9.1,3X,'R(0):',X,F9.1)
   95 FORMAT(/X,'KQ(',I1,'):',X,F9.1,3X,'RQ(',I1,'):',X,F9.1,3X,
     #       'R(',I1,'):',X,F9.1)
      END
```

Betekenis van de in het programma SPLINE gebruikte variabelen

D : dikte van de laag winbaar zand [m]
DATE : datum van de programmarun [-]
EXAMPLE: identificatie van de invoergegevens [-]
K : index van de spline functie intervallen [-]
KQ(K) : zandproduktie waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KU : aantal spline functie intervallen [-]
R(K) : terreinoppervlakte als functie van de zandproduktie RQ(K) [m^2]
RA(K) : afgeleide van de functie R(K) [m^{-1}]
RO : relatieve afwijking $\frac{RS(q) - R(q)}{R(q)} < RO$ [-]
RQ(K) : zandproduktie die correspondeert met het begin van het K-de spline functie interval [m^3]
RQ0 : minimaal te produceren hoeveelheid zand ingeval de winput in produktie genomen wordt [m^3]
T : talud van de winput [-]
V : dikte van de niet uit zand bestaande bovenlaag [m]

Invoerbestand van het rekenvoorbeeld

22-10-1986
Rekenvoorbeeld ICW-Nota 1343, relatieve afwijking RO
15.0,0.01,30000.0,4.0,3.0

Uitvoerbestand van het rekenvoorbeeld

22-10-1986
Rekenvoorbeeld ICW-Nota 1343, relatieve afwijking RO

Gebruikte parameter:

KU: 5

Ingelezen gegevens:

D: 15.0 RO: 0.01 RQ0: 30000.0 T: 4.0 V: 3.0

Berekende variabelen:

		RQ0 :	30000.0	R(0):	12911.3
KQ(1):	46657.1	RQ(1):	72000.0	R(1):	20736.0
KQ(2):	98579.6	RQ(2):	138242.1	R(2):	30113.7
KQ(3):	197890.4	RQ(3):	287128.3	R(3):	47501.0
KQ(4):	428590.5	RQ(4):	644409.6	R(4):	83148.1
KQ(5):	1013685.8	RQ(5):	1600655.1	R(5):	167136.7

BIJLAGE XIX

Listing van het programma TESTRECA

```
      PROGRAM TESTRECA
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,SU
C
C     Definiëren van de parameters
C
      PARAMETER (KU=5,M=5,N=5,SU=25)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),ALFA(N),B(N),CA(N),CB(N),COEF(N,SU,M),CW(N),DT(SU),
     #        DW(N,M),QA,QBU(N,SU)
      REAL AU(N),D(N),DELTA(N),E(N,SU),F(N,M),G(N,M),H(N,KU),KQ(N,KU),
     #     PG(N),Q(N,KU),RQ(N,KU),RQ0(N),T(N),V(N)
C
C     Declareren van de indices
C
      INTEGER I,J,K,S
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*62 AREA
C
C     Communiceren met de subroutine RECALCO
C
      COMMON/BLOCK1/ A,AU,ALFA,B,CA,CB,COEF,CW,D,DELTA,DT,DW,E,F,G,
     #               H,KQ,PG,Q,QA,QBU,RQ,RQ0,T,V
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) AREA
      WRITE(2,25) DATE,AREA
C
C     Aanroepen van de subroutine RECALCO
C
      CALL RECALCO
C
C     Uitvoeren van de gebruikte parameters en ingelezen variabelen
C
      WRITE(2,35)
      WRITE(2,45) KU,M,N,SU
      WRITE(2,55)
      DO 20 I=1,N
            DO 10 J=1,M
                  WRITE(2,65) I,J,(COEF(I,S,J),S=1,SU)
   10       CONTINUE
            WRITE(2,75) I,ALFA(I),I,CA(I),I,CB(I),I,CW(I)
```

```
            WRITE(2,85) I,D(I),I,DELTA(I),I,PG(I),I,RQ0(I)
            WRITE(2,95) I,T(I),I,V(I)
            WRITE(2,105) I,(DW(I,J),J=1,M)
   20 CONTINUE
      WRITE(2,115) (A(J),J=1,M)
      WRITE(2,125) (DT(S),S=1,SU)
C
C     Uitvoeren van de berekende coëfficiënten
C
      WRITE(2,135)
      DO 30 I=1,N
            WRITE(2,145) I,AU(I)
            WRITE(2,155) I,B(I)
            WRITE(2,165) I,(E(I,S),S=1,SU)
            WRITE(2,175) I,(F(I,J),J=1,M)
            WRITE(2,185) I,(G(I,J),J=1,M)
            WRITE(2,195) I,(H(I,K),K=1,KU)
            WRITE(2,205) I,(KQ(I,K),K=1,SU)
            WRITE(2,215) I,(Q(I,K),K=1,KU)
            WRITE(2,225) I,(QBU(I,S),S=1,SU)
   30 CONTINUE
      WRITE(2,235) QA
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A62)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,'Gebruikte parameters:')
   45 FORMAT(/X,'KU:',X,I1,3X,'M:',X,I1,3X,'N:',X,I1,3X,'SU:',X,I2)
   55 FORMAT(//X,'Ingelezen gegevens:')
   65 FORMAT(/X,'COEF(',I1,',S,',I1,'):',25(X,I1))
   75 FORMAT(/X,'ALFA(',I1,')',4X,':',X,I5,3X,'CA(',I1,')',3X,':',
     #       X,I6,3X,'CB(',I1,'):',X,I5,3X,'CW(',I1,')',X,':',3X,I5)
   85 FORMAT(/X,'D(',I1,')',7X,':',2X,F4.1,3X,'DELTA(',I1,'):',X,
     #       F6.4,3X,'PG(',I1,'):',3X,F3.1,3X,'RQ0(',I1,'):',X,F7.1)
   95 FORMAT(/X,'T(',I1,')',7X,':',3X,F3.1,3X,'V(',I1,')',4X,':',4X,
     #       F3.1)
  105 FORMAT(/X,'DW(',I1,'J)',4X,':',5(X,I2))
  115 FORMAT(/X,'A(J)',7X,':',X,I7,4(3X,I7))
  125 FORMAT(/X,'DT(S)',6X,':',20(X,I2)/(13X/20(X,I2)))
  135 FORMAT(//X,'Berekende coëfficiënten:')
  145 FORMAT(/X,'AU(',I1,')',3X,':',2X,F8.1)
  155 FORMAT(/X,'B(',I1,')',4X,':',5X,I5)
  165 FORMAT(/X,'E(',I1,',S)',2X,':',2X,F8.1,4(4X,F8.1)/(8X,5(4X,
     #       F8.1)))
  175 FORMAT(/X,'F(',I1,',J)',2X,':',5X,F5.3,4(7X,F5.3))
  185 FORMAT(/X,'G(',I1,',J)',2X,':',5X,F5.3,4(7X,F5.3))
  195 FORMAT(/X,'H(',I1,',J)',2X,':',4X,F6.3,4(6X,F6.3))
  205 FORMAT(/X,'KQ(',I1,',K)',X,':',X,F9.1,4(3X,F9.1))
  215 FORMAT(/X,'Q(',I1,',K)',2X,':',F10.1,4(2X,F10.1))
  225 FORMAT(/X,'QBU(',I1,',S):',3X,I7,4(5X,I7)/(8X,5(5X,I7)))
  235 FORMAT(/X,'QA',6X,':',3X,I7)
      END
```

Listing van de subroutine RECALCO

```
      SUBROUTINE RECALCO
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,SU
C
C     Definiëren van de parameters
C
      PARAMETER(KU=5,M=5,N=2,SU=25)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),ALFA(N),B(N),CA(N),CB(N),COEF(N,SU,M),CW(N),DT(SU),
     #        DW(N,M),QA,QBU(N,SU)
      REAL AU(N),D(N),DELTA(N),E(N,SU),F(N,M),G(N,M),H(N,KU),
     #     KQ(N,KU),PG(N),Q(N,KU),RQ(N,KU),RQ0(N),T(N),V(N)
C
C     Declareren van de lokale variabelen
C
      REAL DB(N,M),R(N,0:KU),RA(N,0:KU),RO
C
C     Declareren van de indices
C
      INTEGER I,J,K,S
C
C     Communiceren met het programma TESTRECA
C
      COMMON/BLOCK1/ A,AU,ALFA,B,CA,CB,COEF,CW,D,DELTA,DT,DW,E,F,G,H,
     #               KQ,PG,Q,QA,QBU,RQ,RQ0,T,V
C
C     Definiëren van de relatieve afwijking RO
C
      DATA RO /0.01/
C
C     Inlezen van de invoergegevens
C
      READ(1,*) (((COEF(I,S,J),S=1,SU),J=1,M),I=1,N)
      DO 10 I=1,N
            READ(1,*) ALFA(I),CA(I),CB(I),CW(I),D(I),DELTA(I),
     #                PG(I),RQ0(I),T(I),V(I),(DW(I,J),J=1,M)
   10 CONTINUE
      READ(1,*) (A(J),J=1,M)
      READ(1,*) (DT(S),S=1,SU)
C
C     Berekenen van de coëfficiënten
C
      QA=0
      DO 20 J=1,M
            QA=QA+A(J)
   20 CONTINUE
      DO 110 I=1,N
            R(I,0)=(2*T(I)*V(I)+(6*T(I)*RQ0(I))**(1.0/3))**2
            RA(I,0)=4*T(I)*(6*T(I)*RQ0(I))**(-2.0/3)*(2*T(I)*V(I)+
     #              (6*T(I)*RQ0(I))**(1.0/3))
```

```
          AU(I)=PG(I)*(R(I,0)-RQ0(I)*RA(I,0))+CA(I)+CB(I)
          B(I)=CW(I)-CB(I)
          DO 40 S=1,SU
                DO 30 J=1,M
                      IF(COEF(I,S,J).EQ.0)GOTO 30
                      E(I,S)=ALFA(I)*DT(S)/10.0
                      GOTO 40
 30             CONTINUE
                E(I,S)=0.0
 40       CONTINUE
          DO 60 J=1,M
                F(I,J)=PG(I)*RA(I,0)+2.6+0.35*DW(I,J)
                DB(I,J)=0.0
                DO 50 S=1,SU
                      DB(I,J)=DB(I,J))+COEF(I,S,J)*DT(S)/10.0
 50             CONTINUE
                G(I,J)=PG(I)*RA(I,0)+0.66+DELTA(I)*DB(I,J)
 60       CONTINUE
          RQ(I,1)=4.0/3*D(I)**3*T(I)**2
          DO 70 K=1,KU
                R(I,K)=((RQ(I,K)/D(I)-T(I)**2*D(I)**2/3)**0.5+T(I)*
   #                  D(I)+2*T(I)*V(I))**2
                RA(I,K)=(1+(T(I)*D(I)+2*T(I)*V(I))/(RQ(I,K)/D(I)-
   #                   T(I)**2*D(I)**2/3)**0.5)/D(I)
                H(I,K)=PG(I)*(RA(I,K)-RA(I,K-1))
                IF(K.EQ.1)THEN
                  KQ(I,1)=(R(I,1)-R(I,0)+RQ0(I)*RA(I,0)-RQ(I,1)*
   #                     RA(I,1))/(RA(I,0)-RA(I,1))
                ENDIF
                Q(I,K)=QA-KQ(I,K)
                IF(K.EQ.KU)GOTO 80
                KQ(I,K+1)=T(I)**2*D(I)**3/3+D(I)*(((R(I,K)*(D(I)*
   #                     RA(I,K)-1)*RO)**0.5+(1+RO)*(T(I)*D(I)+2*
   #                     T(I)*V(I)))/(D(I)*RA(I,K)-1-RO))**2
                RQ(I,K+1)=T(I)**2*D(I)**3/3+D(I)*(KQ(I,K+1)/D(I)-
   #                     T(I)**2*D(I)**2/3)**2*(RQ(I,K)/D(I)-
   #                     T(I)**2*D(I)**2/3)
 70       CONTINUE
 80       CONTINUE
          DO 100 S=1,SU
                 QBU(I,S)=0
                 DO 90 J=1,M
                       QBU(I,S)=QBU(I,S)+COEF(I,S,J)*A(J)
 90              CONTINUE
100       CONTINUE
110 CONTINUE
    RETURN
    END
```

Betekenis van de in het programma TESTRECA en de subroutine RECALCO gebruikte variabelen

A(J)	: vraag naar zand van vraagpunt J [m^3]
AREA	: identificatie van het proefgebied [-]
AU(I)	: parameter voor de vaste kosten van het in produktie nemen van winput I [f]
ALFA(I)	: kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand van winput I [f/km]
B(I)	: parameter voor de vaste kosten van het transport per vrachtwagen vanuit winput I [f]
CA(I)	: vaste kosten van het in produktie nemen van winput I [f]
CB(I)	: vaste winningskosten van winput I bij transport per pijpleiding [f]
COEF(I,S,J)	: 0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
CW(I)	: vaste winningskosten van winput I bij transport per vrachtwagen [f]
D(I)	: dikte van de laag winbaar zand van winput I [m]
DATE	: datum van de programma-run [-]
DB(I,J)	: lengte van het totale leidingtracé dat winput I met vraagpunt J verbindt [km]
DELTA(I)	: kostencoëfficiënt die afhankelijk is van de gemiddelde korreldiameter van het zand van winput I [f/km]
DT(I,S)	: lengte van traject S van het leidingennet van winput I [hm]
DW(I,J)	: gewogen transportafstand van winput I naar vraagpunt J bij transport per vrachtwagen [km]
E(I,S)	: parameter voor de vaste kosten van traject S van het leidingennet van winput I [f]
F(I,J)	: coëfficiënt voor de variabele kosten van het transport per vrachtwagen van winput I naar vraagpunt J [f/m^3]
G(I,J)	: coëfficiënt voor de variabele kosten van het transport per pijpleiding van winput I naar vraagpunt J [f/m^3]
H(I,K)	: coëfficiënt voor de reductie in variabele kosten wanneer de produktie van winput I een hoeveelheid KQ(I,K) overschrijdt [f/m^3]
I	: index van de winputten [-]
J	: index van de vraagpunten [-]
K	: index van de spline functie intervallen [-]
KQ(I,K)	: zandproduktie van winput I waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KU	: aantal spline functie intervallen [-]
M	: aantal vraagpunten [-]
N	: aantal winputten [-]
PG(I)	: grondprijs van winput I [f/m^2]
Q(I,K)	: QA-KQ(I,K) [m^3]
QA	: totale vraag naar zand [m^3]
QBU(I,S)	: maximale hoeveelheid zand die vanuit winput I getransporteerd wordt door leidingtraject S [m^3]

R(I,K)	:	terreinoppervlakte als functie van de zandproduktie RQ(I,K) van winput I [m^2]
RA(I,K)	:	afgeleide van de functie R(I,K) [m^{-1}]
RO	:	relatieve afwijking $\frac{RS(q)-R(q)}{R(q)} < RO$ [-]
RQ(I,K)	:	zandproduktie van winput I die correspondeert met het begin van het K-de spline functie interval [m^3]
RQ0(I)	:	minimaal te produceren hoeveelheid zand ingeval winput I in produktie genomen wordt [m^3]
S	:	index van de leidingtrajecten [-]
SU	:	aantal leidingtrajecten [-]
T(I)	:	talud van de winput I [-]
V(I)	:	dikte van de niet uit zand bestaande bovenlaag van winput I [m]

Invoerbestand van de subroutinetest

```
22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41
2*1,4*0,3*1,0,2*1,13*0,
12*0,10*1,3*0,
12*0,1,9*0,1,2*0,
12*0,1,9*0,2*1,0,
12*0,1,9*0,3*1,
1,5*0,2*1,17*0,
3*0,3*1,3*0,1,3*0,3*1,9*0,
0,2*1,5*0,1,14*0,1,0,
0,2*1,5*0,1,16*0,
0,2*1,5*0,1,15*0,1,
43850,100000,17000,32000,28.5,0.3355,5.0,50000.0,4.0,2.0,
10,12, 1, 2, 3,
56600,100000,17000,32000,30.0,0.4150,5.0,50000.0,4.0,2.5,
 5,17,12,12,11,
1000000,4*40000,
12,38,24,16,34,11, 9,11,21,43,15,19, 9,11,10,12,10,10,14,19,
15, 6, 6,11,10
```

Uitvoerbestand van de subroutinetest

```
22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41


Gebruikte parameters:

KU: 5   M: 5   N: 2   SU: 25


Ingelezen gegevens:

COEF(1,S,1): 1 1 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

COEF(1,S,2): 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0
```

COEF(1,S,3): 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0

COEF(1,S,4): 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0

COEF(1,S,5): 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1

ALFA(1) : 43850 CA(1) : 100000 CB(1): 17000 CW(1) : 32000

D(1) : 28.5 DELTA(1): 0.3355 PG(1): 5.0 RQ0(1): 50000.0

T(1) : 4.0 V(1) : 2.0

DW(1,J) : 10 12 1 2 3

COEF(2,S,1): 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

COEF(2,S,2): 0 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0

COEF(2,S,3): 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

COEF(2,S,4): 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

COEF(2,S,5): 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

ALFA(2) : 56600 CA(2) : 100000 CB(2): 17000 CW(2) : 32000

D(2) : 30.0 DELTA(2): 0.4150 PG(2): 5.0 RQ0(2): 50000.0

T(2) : 4.0 V(2) : 2.5

DW(2,J) : 5 17 12 12 11

A(J) : 1000000 40000 40000 40000 40000

DT(S) : 12 38 24 16 34 11 9 11 21 43 15 19 9 11 10 12 10 10 14 19
15 6 6 11 10

Berekende coëfficiënten:

AU(1) : 148435.8

B(1) : 15000

E(1,S) :	52620.0	166630.0	0.0	0.0	0.0
	0.0	39465.0	48235.0	92085.0	0.0
	65775.0	83315.0	39465.0	48235.0	43850.0
	52620.0	43850.0	43850.0	61390.0	83315.0
	65775.0	26310.0	26310.0	48235.0	43850.0
F(1,J) :	6.966	7.666	3.816	4.166	4.516
G(1,J) :	5.720	5.418	2.029	2.398	2.734
H(1,K) :	-0.491	-0.066	-0.045	-0.032	-0.022

KQ(1,K) :	164678.5	675775.1	1365149.1	3030709.8	7536518.5
Q(1,K) :	995321.6	484224.9	-205149.1	-1870709.8	-6376518.5
QBU(1,S):	1000000	1000000	0	0	0
	0	1000000	1000000	1000000	0
	1000000	1000000	160000	40000	40000
	40000	40000	40000	40000	40000
	40000	40000	120000	80000	40000
AU(2) :	151989.5				
B(2) :	15000				
E(2,S) :	67920.0	215080.0	135840.0	90560.0	192440.0
	62260.0	50940.0	62260.0	118860.0	243380.0
	0.0	0.0	0.0	62660.0	56600.0
	67920.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	62660.0	56600.0
F(2,J) :	5.245	9.445	7.695	7.695	7.345
G(2,J) :	2.883	7.240	5.456	4.999	5.414
H(2,K) :	-0.533	-0.064	-0.044	-0.031	-0.022
KQ(2,K) :	178120.6	788107.1	1590290.6	3518872.0	8694543.0
Q(2,K) :	981879.4	371892.9	-430290.6	-2358872.0	-7534543.0
QBU(2,S):	1000000	120000	120000	40000	40000
	40000	1000000	1000000	120000	40000
	0	0	0	40000	40000
	40000	0	0	0	0
	0	0	0	40000	40000
QA :	1160000				

Controle d.m.v. handberekening

$R(1,0)=[2*4.0*2.0+(6*4.0*50000.0)^{1/3}]^2=14948.9$

$RA(1,0)=4*4.0*(6*4.0*50000)^{-2/3}*(2*4.0*2.0+(6*4.0*50000.0)^{1/3}=0.1732$

AU(1)=5.0*(14948.9-50000.0*0.1732)+100000+17000=148444.5

B(1)=32000-17000=15000

E(1,1)=43850*12/10.0=52620

F(1,1)=5.0*0.1732+2.6+0.35*10=6.966

DB(1,1)=(12+38+9+11+21+15+19/10.0)=12.5

G(1.1)=5.0*0.1732+0.66+0.3355*12.5=5.720

$$RQ(1,1)=\frac{4.0}{3}*28.5^3*4.0^2=493848.0$$

$$RA(1,1)=\frac{1}{28.5}*\left[1+[4.0*28.5+2*4.0*2.0]/[\frac{493848.0}{28.5}-(\frac{4.0^2*28.5^2}{3})]^{1/2}\right]=0.075$$

H(1,1)=5.0*(0.075-0.1732)=-0.491

Q(1,1)=1160000-164678.5=995321.1 (waarde KQ(1,1) niet gecontroleerd)

QBU(1,1)=1000000+0+0+0+0=1000000

QA=1000000+40000+40000+40000+40000=1160000

BIJLAGE XX

Listing van het programma TESTANSOL

```
      PROGRAM TESTANSOL
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,SU
C
C     Definiëren van de parameters
C
      PARAMETER (KU=5,M=5,N=5,SU=25)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER D1(N),D2(N),D3(N,SU),D4(N,KU)
      REAL TCG
C
C     Declareren van de indices
C
      INTEGER I,K,S
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*62 AREA
C
C     Communiceren met de subroutine ANSOL
C
      COMMON/BLOCK1/ D1,D2,D3,D4,TCG
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) AREA
      WRITE(2,25) DATE,AREA
C
C     Aanroepen van de subroutines RECALCO en ANSOL
C
      CALL RECALCO
      CALL ANSOL
C
C     Uitvoeren van de gebruikte parameters
C
      WRITE(2,35)
      WRITE(2,45) KU,M,N,SU
C
C     Uitvoeren van de berekende variabelen
C
      WRITE(2,55)
      DO 10 I=1,N
            WRITE(2,65) I,D1(I)
            WRITE(2,75) I,D2(I)
            WRITE(2,85) I,(D3(I,S),S=1,SU)
```

```
          WRITE(2,95) I,(D4(I,K),K=1,KU)
   10 CONTINUE
      WRITE(2,105) TCG
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A62)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,'Gebruikte parameters:')
   45 FORMAT(/X,'KU:',X,I1,3X,'M:',X,I1,3X,'N:',X,I1,3X,'SU:',X,I2)
   55 FORMAT(//X,'Berekende variabelen:')
   65 FORMAT(/X,'D1(',I1,')',2X,':',X,I1)
   75 FORMAT(/X,'D2(',I1,')',2X,':',X,I1)
   85 FORMAT(/X,'D3(',I1,',S):',25(X,I1))
   95 FORMAT(/X,'D4(',I1,',K):',5(X,I1))
  105 FORMAT(/X,'TCG',4X,':',X,F9.1)
      END
```

Listing van de subroutine RECALCO

De listing van de subroutine RECALCO is identiek aan die van bijlage XIX met uitzondering van de volgende COMMON-statement:

```
C
C     Communiceren met de subroutine ANSOL
C
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,Q,QA,QBU,RQ0
```

Listing van de subroutine ANSOL

```
    SUBROUTINE ANSOL
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,SU
C
C     Definiëren van de parameters
C
      PARAMETER (KU=5,M=5,N=5,SU=25)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),B(N),COEF(N,SU,M),D1(N),D2(N),D3(N,SU),D4(N,KU),
     #        QA,QBU(N,SU)
      REAL AU(N),E(N,SU),F(N,M),G(N,M),H(N,KU),KQ(N,KU),Q(N,KU),
     #     RQ0(N),TCG
C
C     Declareren van de lokale variabelen
C
      INTEGER II(N),NU(N,M)
      REAL DH1,DH2,DH3,GAM,MIN,PSI1,PSI2,QB(N,M),QHELP,QW(N,M),
```

```
     #     QX(N,KU),SUM1,SUM2,SUM3,ZETA(N,M)
C
C     Declareren van de indices
C
      INTEGER I,J,K,S
C
C     Communiceren met het programma TESTANSOL en de
C     subroutine RECALCO
C
      COMMON/BLOCK1/ D1,D2,D3,D4,TCG
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,Q,QA,QBU,RQ0
C
C     Berekenen van NU(I,J), ZETA(I,J), II(J) en uitvoeren van
C     tussenuitkomsten
C
      WRITE(2,5)
      DO 40 I=1,N
            SUM1=0.0
            DO 10 K=1,KU
                  SUM1=SUM1+H(I,K)*Q(I,K)
   10       CONTINUE
            WRITE(2,15) I
            WRITE(2,25) SUM1
            WRITE(2,35)
            DO 30 J=1,M
                  SUM2=0.0
                  DO 20 S=1,SU
                        SUM2=SUM2+COEF(I,S,J)*E(I,S)/QBU(I,S)
   20             CONTINUE
                  PSI1=F(I,J)+(AU(I)+B(I)+SUM1)/QA
                  PSI2=G(I,J)+(AU(I)+SUM1)/QA+SUM2
                  NU(I,J)=1
                  IF(PSI1.LT.PSI2)NU(I,J)=0
                  ZETA(I,J)=(1-NU(I,J))*PSI1+NU(I,J)*PSI2
                  WRITE(2,45) J,SUM2,PSI1,PSI2,NU(I,J),ZETA(I,J)
   30       CONTINUE
   40 CONTINUE
      WRITE(2,55)
      DO 60 J=1,M
            MIN=ZETA(1,J)
            II(J)=1
            DO 50 I=2,N
                  IF(ZETA(I,J).GE.MIN)GOTO 50
                  MIN=ZETA(I,J)
                  II(J)=I
   50       CONTINUE
            WRITE(2,65) J,II(J)
   60 CONTINUE
C
C     Berekenen van de dummievariabelen X1(J), de minimale kosten TCG
C     en uitvoeren van tussenuitkomsten
C
      TCG=0.0
      DO 140 I=1,N
             WRITE(2,75) I
             D1(I)=0
```

```
            D2(I)=0
            DO 70 S=1,SU
                  D3(I,S)=0
 70         CONTINUE
            SUM3=0.0
            WRITE(2,85)
            DO 80 J=1,M
                  QHELP=0.0
                  IF(II(J).EQ.I)QHELP=A(J)
                  QB(I,J)=NU(I,J)*QHELP
                  QW(I,J)=(1-NU(I,J))*QHELP
                  TCG=TCG+ZETA(I,J)*QHELP
                  SUM3=SUM3+QHELP
                  WRITE(2,95) J,QHELP
 80         CONTINUE
            WRITE(2,105) SUM3
            DH1=SUM3/QA
            IF(DH1.EQ.0.0)GOTO 140
            D1(I)=1
            DH2=0.0
            DO 90 J=1,M
                  DH2=DH2+QW(I,J)/QA
 90         CONTINUE
            IF(DH2.GT.0.0)D2(I)=1
            IF(DH2.EQ.DH1)GOTO 120
            DO 110 S=1,SU
                   DH3=0.0
                   IF(QBU(I,S).EQ.0.0)GOTO 110
                   DO 100 J=1,M
                          DH3=DH3+COEF(I,S,J)*QB(I,J)/QBU(I,S)
100                CONTINUE
                   IF(DH3.GT.0.0)D3(I,S)=1
110         CONTINUE
120         CONTINUE
            GAM=0.0
            IF(SUM3.LT.RQ0(I))GAM=RQ0(I)-SUM3
            WRITE(2,115) GAM
            DO 130 K=1,KU
                   D4(I,K)=1
                   QX(I,K)=SUM3+GAM-KQ(I,K)
                   IF(QX(I,K).LT.0.0)THEN
                     D4(I,K)=0
                     QX(I,K)=0.0
                   ENDIF
130         CONTINUE
            WRITE(2,125) I,(QB(I,J),J=1,M)
            WRITE(2,135) I,(QW(I,J),J=1,M)
            WRITE(2,145) I,(QX(I,K),J=1,KU)
140 CONTINUE
    RETURN
C
C   Format statements
C
  5 FORMAT(/X,'Tussenuitkomsten:')
 15 FORMAT(/X,'I',3X,':',X,I1)
 25 FORMAT(/X,'SUM1:',X,F9.1)
```

```
 35 FORMAT(/X,'J:',6X,'SUM2:',7X,'PSI1:',7X,'PSI2:',5X,'NU:',
   #       6X,'ZETA:'/)
 45 FORMAT(X,I1,3(6X,F6.3),6X,I1,6X,F6.3)
 55 FORMAT(/X,'J:',5X,'II(J):'/)
 65 FORMAT(X,I1,8X,I1)
 75 FORMAT(/X,'I:',X,I1)
 85 FORMAT(/X,'J:',5X,'QHELP:'/)
 95 FORMAT(X,I1,3X,F9.1)
105 FORMAT(/X,'SUM3',3X,':',X,F9.1)
115 FORMAT(/X,'GAM',4X,':',X,F9.1)
125 FORMAT(/X,'QB(',I1,',J):',X,F9.1,4(2X,F9.1))
135 FORMAT(/X,'QW(',I1,',J):',X,F9.1,4(2X,F9.1))
145 FORMAT(/X,'QX(',I1,',J):',X,F9.1,4(2X,F9.1))
    END
```

Betekenis van de in het programma TESTANSOL en de subroutine ANSOL gebruikte variabelen

A(J)	: vraag naar zand van vraagpunt J [m^3]
AREA	: identificatie van het proefgebied [-]
AU(I)	: parameter voor de vaste kosten van het in produktie nemen van winput I [f]
B(I)	: parameter voor de vaste kosten van het transport per vrachtwagen vanuit winput I [f]
COEF(I,S,J)	: 0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
DATE	: datum van de programma-run [-]
DH1,DH2,DH3	: hulpvariabelen [-]
E(I,S)	: parameter voor de vaste kosten van traject S van het leidingennet van winput I [f]
F(I,J)	: coëfficiënt voor de variabele kosten van het transport per vrachtwagen van winput I naar vraagpunt J [f/m^3]
G(I,J)	: coëfficiënt voor de variabele kosten van het transport per pijpleiding van winput I naar vraagpunt J [f/m^3]
GAM	: hulpvariabele [m^3]
H(I,K)	: coëfficiënt voor de reductie in variabele kosten wanneer de produktie van winput I een hoeveelheid KQ(I,K) overschrijdt [f/m^3]
I	: index van de winputten [-]
II(J)	: hulpvariabele [-]
J	: index van de vraagpunten [-]
K	: index van de spline functie intervallen [-]
KQ(I,K)	: zandproduktie van winput I waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KU	: aantal spline functie intervallen [-]
M	: aantal vraagpunten [-]
MIN	: hulpvariabele [f/m^3]
N	: aantal winputten [-]
ND	: totaal aantal dummievariabelen [-]
NU(I,J)	: hulpvariabele [-]
PSI1	: ,, [f/m^3]
PSI2	: ,, [f/m^3]

Q(I,K)	:	QA - KQ(I,K) [m^3]
QA	:	totale vraag naar zand [m^3]
QB(I,J)	:	hoeveelheid zand die per pijpleiding van winput I naar vraagpunt J wordt getransporteerd [m^3]
QBU(I,S)	:	maximale hoeveelheid zand die vanuit winput I getransporteerd wordt door leidingtraject S [m^3]
QHELP	:	hulpvariabele [m^3]
QW(I,J)	:	hoeveelheid zand die per vrachtwagen van winput I naar vraagpunt J wordt getransporteerd [m^3]
QX(I,K)	:	hoeveelheid zand die meer geproduceerd wordt dan de hoeveelheid die correspondeert met de K-de knik in de lineaire spline functie benadering [m^3]
RQ0(I)	:	minimaal te produceren hoeveelheid zand ingeval winput I in produktie genomen wordt [m^3]
S	:	index van de leidingtrajecten [-]
SU	:	aantal leidingtrajecten [-]
SUM1	:	hulpvariabele [f]
SUM2	:	,, [f]
SUM3	:	,, [m^3]
TCG	:	minimale totale kosten [f]
X1(J)	:	dummievariabele [-]
ZETA(I,J)	:	hulpvariabele [f/m^3]

Uitvoerbestand van de subroutinetest

22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41

Gebruikte parameters:

KU: 5 M: 5 N: 2 SU: 25

Tussenuitkomsten:

I : 1

SUM1: -311283.9

J:	SUM2:	PSI1:	PSI2:	NU:	ZETA:
1	0.548	6.869	6.128	1	6.128
2	11.976	7.539	17.254	0	7.539
3	0.466	3.689	2.355	1	2.355
4	1.069	4.039	3.327	1	3.327
5	2.165	4.389	4.759	0	4.389

I : 2

SUM1: -289325.1

J :	SUM2:	PSI1:	PSI2:	NU:	ZETA:
1	0.181	5.140	2.946	1	2.946
2	19.386	9.340	26.508	0	9.340
3	5.481	7.590	10.819	0	7.590
4	3.915	7.590	8.796	0	7.590
5	5.330	7.240	10.626	0	7.240

J:	II(J):
1	2
2	1
3	1
4	1
5	1

I: 1

J:	QHELP:
1	0.0
2	40000.0
3	40000.0
4	40000.0
5	40000.0

SUM3 : 160000.0

GAM : 0.0

QB(1,J):	0.0	0.0	40000.0	40000.0	0.0
QW(1,J):	0.0	40000.0	0.0	0.0	40000.0
QX(1,J):	0.0	0.0	0.0	0.0	0.0

I: 2

J:	QHELP:
1	1000000.0
2	0.0
3	0.0
4	0.0
5	0.0

SUM3 : 1000000.0

GAM : 0.0

QB(2,J):	1000000.0	0.0	0.0	0.0	0.0
QW(2,J):	0.0	0.0	0.0	0.0	0.0
QX(2,J):	821879.4	211892.9	0.0	0.0	0.0

Berekende variabelen:

D1(1) : 1

D2(1) : 1

D3(1,S): 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0

D4(1,K): 0 0 0 0 0

D1(2) : 1

D2(2) : 0

D3(2,S): 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

D4(2,K): 1 1 0 0 0

TCG : 3647040.3

Controle d.m.v. handberekening

I=1:

SUM1=-0.491*995321.6+-0.066*484224.9+-0.045*-205149.1+
-0.032*-1870709.8+-0.022*-6376518.5=-311283.9

J=1:

SUM2=52620.0/1000000+166630.0/1000000+39465.0/1000000+48235.0/1000000+
92085.0/1000000+65775.0/1000000+83315.0/1000000=0.548
PSI1=6.966+(148435.8+15000-311283.9)/1160000=6.996-0.127=6.869

PSI2=5.720+(148435.8-311283.9)/1160000+0.548=5.720-0.140+0.548=6.128

NU=1, ZETA=6.128

ZETA(1,J):	ZETA(2,J):	II(J):
6.128	2.946	2
7.539	9.340	1
2.355	7.590	1
3.327	7.590	1
4.389	7.240	1

I=1:

SUM3=0.0+40000.0+40000.0+40000.0+40000.0=160000.0

GAM=50000.0-160000.0<0.0: GAM=0.0

QB(1,1)=1* 0.0= 0.0 QW(1,1)=(1-1)* 0.0= 0.0
QB(1,2)=0*40000.0= 0.0 QW(1,2)=(1-0)*40000.0=40000.0
QB(1,3)=1*40000.0=40000.0 QW(1,3)=(1-1)*40000.0= 0.0
QB(1,4)=1*40000.0=40000.0 QW(1,2)=(1-1)*40000.0= 0.0
QB(1,5)=0*40000.0= 0.0 QW(1,2)=(1-0)*40000.0=40000.0

QX(1,1)=160000.0-164678.5<0: QX(1,1)=0.0
QX(2,1)=1000000.0- 178120.6=821879.4
QX(2,2)=1000000.0- 788107.1=211892.9
QX(2,2)=1000000.0-1590290.6<0.0: QX)2,3)=0.0

I=1:

DH1=160000.0/11600000>0.0: D1(1)=1

DH2=(0.0+40000.0+0.0+0.0+40000.0)/1160000>0.0: D2(1)=1

S=1: QBU(1,1)=1000000>0:

DH3=(1*0.0+0*0.0+0*40000.0+0*40000.0+0*0.0)/1000000=0.0: D3(1,1)=0

S=13: QBU(1,13)=160000>0:

DH3=(0*0.0+1*0.0+1*40000.0+1*40000.0+1*0.0)/160000>0.0: D3(1,13)=1

QX(1,1)=0.0: D4(1,1)=0
QX(2,1)=821879.4>0.0: D4(2,1)=1

TCG=2.946*1000000.0(7.539+2.355+3.327+4.389)*40000.0=3650400≈3647040.3

Het verschil van 0.09 % is hierbij te wijten aan afrondingsfouten van de bij de handberekening gebruikte waarden.

BIJLAGE XXI

Listing van het programma TESTRESIM

```
      PROGRAM TESTRESIM
C
C     Declareren van de parameters
C
      INTEGER NC,NV
C
C     Definiëren van de parameters
C
      PARAMETER (NC=3,NV=5)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(NC,NV),XB(NC)
      REAL B(NC),BI(NC,NC),C(NV),X(NV),Z
C
C     Declareren van de indices
C
      INTEGER I,J
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*14 PROBLEM
C
C     Communiceren met de subroutine RESIM
C
      COMMON/BLOCK1/ A,B,BI,C,X,XB,Z
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) PROBLEM
      WRITE(2,25) DATE,PROBLEM
C
C     Inlezen van de invoergegevens
C
      READ(1,*) ((A(I,J),J=1,NV),I=1,NC)
      READ(1,*) (B(I),I=1,NC)
      READ(1,*) ((BI(I,J),J=1,NC),I=1,NC)
      READ(1,*) (C(J),J=1,NV)
      READ(1,*) (XB(I),I=1,NC)
C
C     Uitvoeren van de gebruikte parameters en ingelezen gegevens
C
      WRITE(2,35)
      WRITE(2,45) NC,NV
      WRITE(2,55)
      WRITE(2,65) ((A(I,J),J=1,NV),I=1,NC)
      WRITE(2,75) (B(I),I=1,NC)
      WRITE(2,85) ((BI(I,J),J=1,NC),I=1,NC)
      WRITE(2,95) (C(J),J=1,NV)
```

```
      WRITE(2,105) (XB(I),I=1,NC)
C
C     Aanroepen van de subroutine RESIM
C
      CALL RESIM(*20,*10)
C
C     Uitvoeren van de berekende variabelen
C
   10 CONTINUE
      WRITE(2,115)
      WRITE(2,125) (XB(I),I=1,NC)
      WRITE(2,135) (X(J),J=1,NV)
      WRITE(2,145) Z
   20 CONTINUE
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A14)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,'Gebruikte parameters:')
   45 FORMAT(/X,'NC:',X,I2,3X,'NV:',X,I2)
   55 FORMAT(//X,'Ingelezen gegevens:')
   65 FORMAT(/X,'A(I,J)',X,':',X,5(3X,I2)/(10X,5(3X,I2)))
   75 FORMAT(/X,'B(I)',3X,':',X,3(X,F4.1))
   85 FORMAT(/X,'BI(I,J):',X,3(X,F4.1)/(10X,3(X,F4.1)))
   95 FORMAT(/X,'C(J)',3X,':',X,5(X,F4.1))
  105 FORMAT(/X,'XB(I)',2X,':',X,3(3X,I2))
  115 FORMAT(//X,'Berekende variabelen:')
  125 FORMAT(/X,'XB(I)',2X,':',X,3(3X,I2))
  135 FORMAT(/X,'X(J)',3X,':',X,5(X,F4.1))
  145 FORMAT(/X,'Z',6X,':',2X,F4.1)
      END
```

Listing van de subroutine RESIM

```
      SUBROUTINE RESIM(*,*)
C
C     Declareren van de parameters
C
      INTEGER NC,NV
C
C     Definiëren van de parameters
C
      PARAMETER (NC=3,NV=5)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(NC,NV),XB(NC)
      REAL B(NC),BI(NC,NC),C(NV),X(NV),Z
C
C     Declareren van de lokale variabelen
C
      INTEGER IND,K,R
```

```
      REAL AA(NC),BA,FRAC,HELP,HELP1,HELP2(NC),PROD(NC),SUM
C
C     Declareren van de indices
C
      INTEGER I,J
C
C     Communiceren met het programma TESTRESIM
C
      COMMON/BLOCK1/ A,B,BI,C,X,XB,Z
C
C     Controleren of de startbasis toelaatbaar is (BI*B ≥ 0); zo niet,
C     dan terugkeren naar het programma TESTRESIM en de uitvoering
C     ervan onderbreken
C
      DO 20 I=1,NC
            HELP=0.0
            DO 10 J=1,NC
                  HELP=HELP+BI(I,J)*B(J)
   10       CONTINUE
            IF(HELP.LT.0.0)THEN
              WRITE(2,5)
              RETURN 1
            ENDIF
   20 CONTINUE
C
C     Stap 1
C
   30 CONTINUE
      K=0
      DO 50 J=1,NC
            PROD(J)=0.0
            DO 40 I=1,NC
                  PROD(J)=PROD(J)+C(XB(I))*BI(I,J)
   40       CONTINUE
   50 CONTINUE
      HELP=0.0
      DO 70 J=1,NV
            SUM=0.0
            DO 60 I=1,NC
                  SUM=SUM+PROD(I)*A(I,J)
   60       CONTINUE
            SUM=SUM-C(J)
            IF(HELP.GE.SUM)GOTO 70
            HELP=SUM
            K=J
   70 CONTINUE
      IF(K.EQ.0)GOTO 160
C
C     Stap 2
C
      IND=1
      HELP=0.0
      R=0
      DO 100 I=1,NC
             AA(I)=0.0
             BA=0.0
```

```
            DO 80 J=1,NC
                  AA(I)=AA(I)+BI(I,J)*A(J,K)
                  BA=BA+BI(I,J)*B(J)
   80       CONTINUE
            IF(AA(I).LE.0.0)GOTO 100
            IF(IND.EQ.0)GOTO 90
            FRAC=BA/AA(I)
            IF(FRAC.LT.HELP)GOTO 100
            R=I
            IND=0
            GOTO 100
   90       CONTINUE
            HELP=BA/AA(I)
            IF(FRAC.LT.HELP)GOTO 100
            FRAC=HELP
            R=I
  100 CONTINUE
C
C     Controleren of een variabele gevonden kan worden die de basis
C     verlaat (R>0); zo niet, dan is één van de variabelen onbegrensd
C     en kan geen eindige oplossing gevonden worden; terugkeren naar
C     het programma TESTRESIM en de uitvoering ervan onderbreken
C
      IF(R.EQ.0)THEN
        WRITE(2,15)
        RETURN 1
      ENDIF
C
C     Stap 3
C
      HELP1=AA(R)
      DO 110 I=1,NC
            AA(I)=-AA(I)/HELP1
  110 CONTINUE
      AA(R)=1.0/HELP1
      DO 120 J=1,NC
            HELP2(J)=BI(R,J)
  120 CONTINUE
      DO 140 I=1,NC
            IF(I.EQ.R)GOTO 140
            DO 130 J=1,NC
                  BI(I,J)=BI(I,J)+AA(I)*HELP2(J)
  130       CONTINUE
  140 CONTINUE
      I=R
      DO 150 J=1,NC
            BI(I,J)=AA(R)*HELP2(J)
  150 CONTINUE
C
C     Stap 4
C
      GOTO 30
C
C     Berekenen van de variabelen X(J) en de waarde van de doelstel-
C     lingsfunctie Z
C
```

```
  160 CONTINUE
      DO 170 J=1,NV
             X(J)=0.0
  170 CONTINUE
      DO 190 I=1,NC
             DO 180 J=1,NC
                    X(XB(I))=X(XB(I))+B(I,J)*B(J)
  180        CONTINUE
  190 CONTINUE
      DO 200 J=1,NV
             X(J)=0.0
  200 CONTINUE
      DO 220 I=1,NC
             DO 210 J=1,NV
                    X(XB(I)=X(XB(I)+BI(I)*B(J)
  210        CONTINUE
  220 CONTINUE
      Z=0.0
      DO 230 J=1,NV
             Z=Z+C(J)*X(J)
  230 CONTINUE
      RETURN 2
C
C     Format statements
C
    5 FORMAT(//X,'Het programma TESTRESIM is gestopt omdat de ',
     #       'startbasis'/X,'niet toelaatbaar is')
   15 FORMAT(//X,'Het programma TESTRESIM is gestopt omdat één ',
     #       'van de variabelen'/X,'onbegrensd is')
      END
```

Betekenis van de in het programma TESTRESIM en de subroutine RESIM gebruikte variabelen

```
A(I,J)   : coëfficiëntenmatrix van de beperkingen [-]
AA(I)    : hulpvariabele [-]
B(I)     : constante van de beperkingen [-]
BA       : hulpvariabele [-]
BI(I,J)  : inverse van de coëfficiëntenmatrix van de basisvaria-
           belen [-]
C(J)     : coëfficiënt van de variabelen in de doelstellings-
           functie [-]
DATE     : datum van de programma-run [-]
FRAC     : hulpvariabele [-]
HELP     :       ,,      [-]
HELP1    :       ,,      [-]
HELP2(J) :       ,,      [-]
I        : index van de beperkingen [-]
IND      : hulpvariabele [-]
J        : index van de variabelen [-]
K        : index die aangeeft welke variabele basisvariabele
           wordt [-]
NC       : aantal beperkingen van het probleem [-]
NV       : aantal variabelen van het probleem [-]
PROBLEM  : identificatie van het probleem [-]
```

```
PROD(J) : hulpvariabele [-]
R       : index die aangeeft welke variabele de basis verlaat [-]
SUM     : hulpvariabele [-]
X(J)    : continue variabele [-]
XB(I)   : variabele die aangeeft welke variabelen basisvariabele
          zijn [-]
Z       : minimum [-]
```

Invoerbestand van testprobleem 1

```
22-10-1986
Testprobleem 1
   1,    0,    1,    0,    0,
   0,    1,    0,    1,    0,
   3,    2,    0,    0,    1,
 4.0,  6.0, 18.0,
 1.0,  0.0,  0.0,
 0.0,  1.0,  0.0,
 0.0,  0.0,  1.0,
-3.0, -5.0,  0.0,  0.0,  0.0,
   3,    4,    5
```

Uitvoerbestand van testprobleem 1

```
22-10-1986
Testprobleem 1


Gebruikte parameters:

NC:  3   NV:  5


Ingelezen gegevens:

A(I,J) :     1    0    1    0    0
             0    1    0    1    0
             3    2    0    0    1

B(I)   :   4.0  6.0 18.0

BI(I,J):   1.0  0.0  0.0
           0.0  1.0  0.0
           0.0  0.0  1.0

C(J)   :  -3.0 -5.0  0.0  0.0  0.0

XB(I)  :     3    4    5

Berekende variabelen:

XB(I)        3    2    1

X(J)   :   2.0  6.0  2.0  0.0  0.0

Z      : -36.0
```

Invoerbestand van testprobleem 2

```
22-10-1986
Testprobleem 2
   1,   1,   1,   0,   0,
   1,  -1,   0,   1,   0,
   1,   0,   0,   0,   1,
 4.0, 0.0, 1.0,
 1.0, 0.0, 0.0,
 0.0, 1.0, 0.0,
 0.0, 0.0, 1.0,
 5.0,-3.0, 0.0, 0.0, 0.0,
   3,   4,   5
```

Uitvoerbestand van testprobleem 2

```
22-10-1986
Testprobleem 2


Gebruikte parameters:

NC:  3   NV:  5


Ingelezen gegevens:

A(I,J) :      1    1    1    0    0
              1   -1    0    1    0
              1    0    0    0    1

B(I)   :    4.0  0.0  1.0

BI(I,J):    1.0  0.0  0.0
            0.0  1.0  0.0
            0.0  0.0  1.0

C(J)   :    5.0 -3.0  0.0  0.0  0.0

XB(I)  :      3    4    5


Berekende variabelen:

XB(I)  :      2    4    5

X(J)   :    0.0  4.0  0.0  4.0  1.0

Z      : -12.0
```

Invoerbestand van testprobleem 3

```
22-10-1986
Testprobleem 3
   1,    1,   -1,    0,    0,
   1,   -1,    0,   -1,    0,
   1,    0,    0,    0,   -1,
 4.0,  0.0,  1.0,
-1.0,  0.0,  0.0,
 0.0,- 1.0,  0.0,
 0.0,  0.0, -1.0,
 5.0, -3.0,  0.0,  0.0,  0.0,
   3,    4,    5
```

Uitvoerbestand van testprobleem 3

```
22-10-1986
Testprobleem 3


Gebruikte parameters:

NC:  3   NV:  5


Ingelezen gegevens:

A(I,J) :      1    1   -1    0    0
              1   -1    0   -1    0
              1    0    0    0   -1

B(I)   :    4.0  0.0  1.0

BI(I,J):    1.0  0.0  0.0
            1.0 -1.0  0.0
            1.0  0.0 -1.0

C(J)   :    5.0 -3.0  0.0  0.0  0.0

XB(I)  :      1    4    5


Berekende variabelen:

XB(I)   :     1    2    5

X(J)    :   2.0  2.0  0.0  0.0  1.0

Z       :   4.0
```

BIJLAGE XXII

Listing van het programma TESTPRISOL

```
      PROGRAM TESTPRISOL
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,NCP,NTP,NVP,SU
C
C     Definiëren van de parameters
C
      PARAMETER (KU=5,M=5,N=2,SU=86)
      PARAMETER (NCP=N+M,NTP=2*N*M+N+M+N*KU,NVP=N*M+N+M)
C
C     Declareren van de niet-lokale variabelen
C
      REAL PRIOPT,RES,X(NTP),ZR
C
C     Declareren van de indices
C
      INTEGER J
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*62 AREA
C
C     Communiceren met de subroutines PRISOL en RESIM
C
      COMMON/BLOCK1/ PRIOPT,RES,X
      COMMON/BLOCK2/ ZR
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) AREA
      WRITE(2,25) DATE,AREA
C
C     Uitvoeren van de gebruikte parameters
C
      WRITE(2,35)
      WRITE(2,45) KU,M,N,SU
      WRITE(2,55) NCP,NTP,NVP
C
C     Aanroepen van de subroutines RECALCO, ANSOL en PRISOL
C
      CALL RECALCO
      CALL ANSOL
      CALL PRISOL(*10,*20)
C
C     Uitvoeren van de berekende variabelen
C
   10 CONTINUE
      WRITE(2,65)
```

```
      WRITE(2,75) ZR
      WRITE(2,85) (X(J),J=1,NTP)
      WRITE(2,95) RES
      WRITE(2,105) PRIOPT
   20 CONTINUE
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A62)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,'Gebruikte parameters:')
   45 FORMAT(/X,'KU',X,':',X,I2,3X,'M',2X,':',X,I2,3X,'N',2X,':',X,I2,
     #       3X,'SU:',X,I2)
   55 FORMAT(/X,'NCP:',X,I2,3X,'NTP:',X,I2,3X,'NVP:',X,I2)
   65 FORMAT(//X,'Berekende variabelen:')
   75 FORMAT(/X,'ZR',4X,':',X,F9.1)
   85 FORMAT(/X,'X(J)',2X,':',X,F9.1,4(3X,F9.1)/(6X,5(3X,F9.1)))
   95 FORMAT(/X,'RES',3X,':',X,F9.1)
  105 FORMAT(/X,'PRIOPT:',X,F9.1)
      END
```

Listing van de subroutine RECALCO

De listing van de subroutine RECALCO is identiek aan die van bijlage XV met uitzondering van de volgende COMMON-statements:

```
C
C     Communiceren met de subroutines ANSOL en PRISOL
C
      COMMON/BLOCK3/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK4/ Q,QA,QBU
```

Listing van de subroutine ANSOL

De listing van de subroutine ANSOL is identiek aan die van bijlage XVI zonder het uitvoeren van de tussenuitkomsten en met uitzondering van de volgende COMMON-statements:

```
C
C     Communiceren met de subroutines RECALCO en PRISOL
C
      COMMON/BLOCK3/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK4/ Q,QA,QBU
      COMMON/BLOCK5/ D1,D2,D3,D4
```

Listing van de subroutine PRISOL

```
      SUBROUTINE PRISOL(*,*)
C
C     Declareren van de parameters
C
```

```
      INTEGER KU,M,N,NCP,ND,NTP,NVP,SU
C
C     Definiëren van de parameters
C
      PARAMETER (KU=5,M=5,N=2,SU=86)
      PARAMETER (NCP=N+M,NTP=2*N*M+N+M+N*KU,NVP=N*M+N+M)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),AM(NCP,NVP),B(N),COEF(N,SU,M),D1(N),D2(N),D3(N,SU),
     #        D4(N,KU),NC,NV,XB(NCP),
     #
      REAL AU(N),BI(NCP,NCP),BL(NCP),C(NVP),E(N,SU),F(N,M),G(N,M),
     #     H(N,KU),KQ(N,KU),PRIOPT,RES,RQ0(N),X(NTP),XR(NVP),
     #     ZR
C
C     Declareren van de lokale variabelen
C
      INTEGER ETA(N,M),II(M),IND,IO(N),KS,NIO
      REAL HELP1,HELP2,LARGE,MIN,PSI(N,M),SUM,SUM1,SUM2
C
C     Declareren van de indices
C
      INTEGER I,IH,IH1,IH2,J,JH,JH1,JH2,K,S
C
C     Communiceren met het programma TESTPRISOL en de subroutines
C     RECALCO, ANSOL en RESIM
C
      COMMON/BLOCK1/ PRIOPT,RES,X
      COMMON/BLOCK2/ ZR
      COMMON/BLOCK3/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK5/ D1,D2,D3,D4
      COMMON/BLOCK6/ AM,BL,BI,C,NC,NV,XB,XR
C
C     Bepalen van de right hand side BL(J) en uitvoeren van tussen-
C     uitkomsten
C
      DO 10 J=1,M
            BL(J)=A(J)
   10 CONTINUE
      NIO=0
      LARGE=99.999
      WRITE(2,5)
      DO 30 I=1,N
            IF(D1(N).EQ.0)GOTO 30
            NIO=NIO+1
            IO(NIO)=I
            BL(M+NIO)=RQ0(I)
            KS=0
            DO 20 K=1,KU
                  KS=KS+D4(I,K)
   20       CONTINUE
            WRITE(2,15) I,KS
            IF(KS.NE.0)BL(M+NIO)=KQ(I,KS)
   30 CONTINUE
      WRITE(2,25) NIO,(I,IO(I),I=1,NIO)
```

```
      WRITE(2,35) BL(I),I=1,M+NIO
C
C     Berekenen van PSI(I,J), C(J) en uitvoeren van tussenuitkomsten
C
      DO 90 IH=1,NIO
            I=IO(IH)
            SUM=0.0
            DO 40 K=1,KU
                  SUM=SUM+D4(I,K)*H(I,K)
   40       CONTINUE
            WRITE(2,45) I,SUM
            DO 80 J=1,M
                  ETA(I,J)=0
                  IF(D1(I).EQ.0)GOTO 50
                  IF(F(I,J).LT.G(I,J)THEN
                    ETA(I,J)=1
                    PSI(I,J)=F(I,J)+SUM
                    GOTO 70
                  ENDIF
   50             CONTINUE
                  DO 60 S=1,SU
                        IF(COEF(I,S,J).EQ.0)GOTO 60
                        IF(D3(I,S).EQ.1)GOTO 60
                        ETA(I,J)=D2(I)
                        PSI(I,J)=ETA(I,J)*(F(I,J)+SUM)+(1-ETA(I,J))*
     #                          LARGE
                        GOTO 70
   60             CONTINUE
                  ETA(I,J)=2
                  PSI(I,J)=G(I,J)+SUM
   70             CONTINUE
                  C(IH-1)*M+J)=PSI(I,J)
   80       CONTINUE
   90 CONTINUE
      DO 100 J=NIO*M+1,NIO*M+M+NIO
            C(J)=0.0
  100 CONTINUE
      WRITE(2,55) ((ETA(I,J),J=1,M),I=1,N)
      WRITE(2,65) ((PSI(I,J),J=1,M),I=1,NIO)
      WRITE(2,75) (C(J),J=1,NIO*M+M+NIO)
C
C     Berekenen van NC en NV en uitvoeren van tussenuitkomsten
C
      NC=M+NIO
      NV=NIO*M+NC
      WRITE(2,85) NC,NV
C
C     Bepalen van AM(I,J) en uitvoeren van tussenuitkomsten
C
      DO 120 I=1,NC
            DO 110 J=1,NV
                  AM(I,J)=0
  110       CONTINUE
  120 CONTINUE
      DO 140 I=1,M
            DO 130 JH=1,NIO
```

```
                    J=(JH-1)*M+I
                    AM(I,J)=1
130         CONTINUE
140 CONTINUE
    DO 160 I=M+1,NC
            DO 150 JH=1,M
                    J=(I-M-1)*M+JH
                    AM(I,J)=1
150         CONTINUE
160 CONTINUE
    DO 170 I=1,NC
            J=NIO*M+I
            AM(I,J)=-1
170 CONTINUE
    WRITE(2,95) ((AM(I,J),J=1,NV),I=1,NC)
C
C     Bepalen van de startbasis (XB(J)) en uitvoeren van tussen-
C     uitkomsten
C
    DO 180 J=1,NV
            X(J)=0.0
180 CONTINUE
    DO 200 J=1,M
            MIN=LARGE
            DO 190 IH=1,NIO
                    I=IO(IH)
                    IF(PSI(I,J).GE.MIN)GOTO 190
                    MIN=PSI(I,J)
                    II(J)=IH
190         CONTINUE
            WRITE(2,105) J,II(J)
200 CONTINUE
    DO 210 J=1,M-1
            XB(J)=(II(J)-1)*M+J
            JH=NIO*M+M+II(J)
            X(JH)=X(JH)+BL(J)
            WRITE(2,115) JH,X(JH)
210 CONTINUE
    DO 220 J=M+1,NC-1
            JH=NIO*M+J
            IF(X(JH).GT.BL(J))THEN
              XB(J)=JH
            ELSE
              XB(J)=(J-M)*M
              X(NIO*M+M)=X(NIO*M+M)+BL(J)
            ENDIF
220 CONTINUE
    JH=NIO*M+M
    SUM1=BL(M)-X(JH)
    SUM2=BL(NC)-X(NV)
    WRITE(2,115) JH,X(JH),NV,X(NV)
    WRITE(2,125) SUM1,SUM2
    IND=0
    XB(M)=NIO*M
    XB(NC)=NIO*M+M
    IF(SUM2.LE.SUM1)THEN
```

```
        IND=1
        XB(NC)=NV
      ENDIF
      IF(IND*SUM1+(1-IND)*SUM2).LE.0.0)THEN
        IND=2
        XB(M)=NIO*M+M
        XB(NC)=NV
      ENDIF
      WRITE(2,135) IND
      WRITE(2,145) (XB(I),I=1,NC)
C
C     Bepalen van de inverse matrix BI(I,J) en uitvoeren van tussen-
C     uitkomsten
C
      DO 240 J=1,NC
             DO 230 I=1,NC
                    BI(I,J)=AM(I,XB(J))
  230        CONTINUE
  240 CONTINUE
      WRITE(2,155) ((BI(I,J),J=1,NC),I=1,NC)
      DO 260 I=1,NIO-1
             DO 250 J=1,M-1
                    HELP1=BI(M+I,J)
                    BI(M+I,J)=-BI(M+I,M+I)*HELP1
                    BI(M,J)=BI(M,J)+HELP1*BI(M,M+I)
  250        CONTINUE
  260 CONTINUE
      IF(IND.EQ.0)THEN
        DO 270 J=1,M-1
               HELP1=BI(M,J)
               HELP2=BI(NC,J)
               BI(M,J)=-HELP2
               BI(NC,J)=-HELP1-HELP2
  270   CONTINUE
        DO 280 I=M+1,NC-1
               HELP1=BI(M,I)
               BI(M,I)=0.0
               BI(NC,I)=HELP1
  280   CONTINUE
        BI(M,M)=0.0
        BI(M,NC)=1.0
        BI(NC,M)=-1.0
        BI(NC,NC)=1.0
      ENDIF
      IF(IND.EQ.1)THEN
        DO 290 J=1,M-1
               HELP1=BI(M,J)
               BI(NC,J)=BI(NC,J)+HELP1
  290   CONTINUE
        DO 300 I=M+1,NC-1
               BI(M,I)=-BI(M,I)
               BI(NC,I)=BI(M,I)
  300   CONTINUE
        BI(M,M)=1.0
        BI(M,NC)=0.0
        BI(NC,M)=1.0
```

```
        BI(NC,NC)=-1.0
      ENDIF
      IF(IND.EQ.2)THEN
         DO 310 J=1,M-1
                BI(M,J)=-BI(M,J)
  310   CONTINUE
        BI(M,M)=-1.0
        BI(M,NC)=0.0
        BI(NC,M)=0.0
        BI(NC,NC)=-1.0
      ENDIF
      WRITE(2,135) IND
      WRITE(2,155) ((BI(I,J),J=1,NC),I=1,NC)
C
C     Aanroepen van de subroutine RESIM
C
      CALL RESIM(*320,*330)
  320 RETURN 2
  330 CONTINUE
C
C     Berekenen van het minimum PRIOPT, de continue variabelen X(J)
C     en uitvoeren van tussenuitkomsten
C
      RES=0.0
      DO 360 I=1,N
             RES=RES+AU(I)*D1(I)+B(I)*D2(I)
             DO 340 S=1,SU
                    RES=RES+E(I,S)*D3(I,S)
  340        CONTINUE
             DO 350 K=1,KU
                    RES=RES-D4(I,K)*H(I,K)*KQ(I,K)
  350        CONTINUE
  360 CONTINUE
      PRIOPT=ZR+RES
      WRITE(2,145) (XB(I),I=1,NC)
      WRITE(2,165) (XR(J),J=1,NV)
      DO 370 J=1,NV
             X(J)=XR(J)
  370 CONTINUE
      DO 380 J=NV+1,NTP
             X(J)=0.0
  380 CONTINUE
      DO 390 J=1,M
             JH1=2*N*M+N*KU+J
             JH2=NIO*M+J
             X(JH1)=X(JH2)
             X(JH2)=0.0
  390 CONTINUE
      DO 400 I=1,NIO
             IH1=2*N*M+N*KU+M+IO(I)
             IH2=NIO*M+M+I
             X(IH1)=X(IH2)
             X(IH2)=0.0
  400 CONTINUE
      DO 420 IH=1,NIO
             I=IO(IH)
```

```
            DO 410 J=1,M
                   JH1=N*M+(I-1)*M+J
                   JH2=(IH-1)*M+J
                   X(JH1)=X(JH2)
                   X(JH2)=0.0
410         CONTINUE
420 CONTINUE
    DO 450 IH=1,NIO
           I=IO(IH)
           SUM=0.0
           DO 430 J=1,M
                  JH1=N*M+(I-1)*M+J
                  IF(X(JH1).EQ.0.0)GOTO 430
                  SUM=SUM+X(JH1)
                  IF(ETA(I,J).EQ.2)GOTO 430
                  JH2=JH1-N*M
                  X(JH2)=X(JH1)
                  X(JH1)=0.0
430        CONTINUE
           DO 440 K=1,KU
                  IF(D4(I,K).EQ.0)GOTO 450
                  J=2*N*M+(I-1)*KU+K
                  X(J)=SUM-KQ(I,K)
440        CONTINUE
450 CONTINUE
    RETURN 1
C
C   Format statements
C
  5 FORMAT(//X,'Tussenuitkomsten:')
 15 FORMAT(/X,'I',7X,':',X,I1,5X,'KS',3X,':',X,I1)
 25 FORMAT(/X,'NIO',5X,':',X,I1,5X,'IO(',I1,'):'X,I1,5X,'IO(',
   #        I1,'):'X,I1)
 35 FORMAT(/X,'BL(I)',3X,':',X,F9.1,4(3X,F9.1)/8X,5(3X,F9.1))
 45 FORMAT(/X,'I',7X,':',X,I1,5X,'SUM:',X,F6.1)
 55 FORMAT(/X,'ETA(I,J):',5(X,I1)/10X,5(X,I1))
 65 FORMAT(/X,'PSI(I,J):',X,F6.3,4(3X,F6.3)/8X,5(3X,F6.3))
 75 FORMAT(/X,'C(J)',4X,':',X,F6.3,4(3X,F6.3)/(8X,5(3X,F6.3)))
 85 FORMAT(/X,'NC',6X,':',X,I2,5X,'NV:',X,I2)
 95 FORMAT(/X,'AM(I,J)',X,':',17(X,I2)/(10X,17(X,I2)))
105 FORMAT(/X,'II(',I1,')',3X,':',X,I1)
115 FORMAT(/X,'X(',I2,')',3X,':',X,F9.1,5X,'X(',I2,'):',X,F9.1)
125 FORMAT(/X,'SUM1',4X,':',X,F9.1,5X,'SUM2',X,':',X,F9.1)
135 FORMAT(/X,'IND',5X,':',X,I1)
145 FORMAT(/X,'XB(I)',3X,':',7(X,I2))
155 FORMAT(/X,'BI(I,J)',X,':',7(X,F4.1)/(10X,7(X,F4.1)))
165 FORMAT(/X,'XR(I)',3X,':',X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
    END
```

Listing van de subroutine RESIM

De listing van de subroutine RESIM is identiek aan die van bijlage XXI met uitzondering van de volgende PARAMETER-, declaratie-, en COMMON-statements:

```
      PARAMETER(M=5,N=2)
      PARAMETER(NCP=N+M,NVP=N*M+N+M)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(NCP,NVP),XB(NCP)
      REAL B(NCP),BI(NCP,NCP),C(NVP),X(NVP),Z
C
C     Declareren van de lokale variabelen
C
      INTEGER IND,K,R
      REAL AA(NCP),BA,FRAC,HELP,HELP1,HELP2(NCP),PROD(NCP),SUM
C
C     Communiceren met de subroutine PRISOL
C
      COMMON/BLOCK2/ Z
      COMMON/BLOCK6/ A,B,BI,C,NC,NV,XB,X
```

Betekenis van de in het programma TESTPRISOL en de subroutine PRISOL gebruikte variabelen

A(J)	: vraag naar zand van vraagpunt J [m^3]
AM(I,J)	: coëfficiëntenmatrix van de beperkingen [-]
AREA	: identificatie van het proefgebied [-]
AU(I)	: parameter voor de vaste kosten van het in produktie nemen van winput I [f]
B(I)	: parameter voor de vaste kosten van het transport per vrachtwagen vanuit winput I [f]
BI(I,J)	: inverse van de coëfficiëntenmatrix van de basis-variabelen [-]
BL(I)	: right hand side van de beperkingen [m^3]
C(J)	: coëfficiënt van de variabelen in de doelstellings-functie [f/m^3]
COEF(I,S,J)	: 0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
DATE	: datum van de programma-run [-]
E(I,S)	: parameter voor de vaste kosten van traject S van het leidingennet van winput I [f]
ETA(I,J)	: hulpvariabele [-]
F(I,J)	: coëfficiënt voor de variabele kosten van het transport per vrachtwagen van winput I naar vraag-punt J [f/m^3]
G(I,J)	: coëfficiënt voor de variabele kosten van het transport per pijpleiding van winput I naar vraag-punt J [f/m^3]
H(I,K)	: coëfficiënt voor de reductie in variabele kosten wanneer de produktie van winput I een hoeveelheid KQ(I,K) overschrijdt [f/m3]
HELP1	: hulpvariabele [-]
HELP2	: ,, [-]
I	: index van de winputten [-]
IH,IH1,IH2	: hulpindices [-]
II(J)	: hulpvariabele [-]
IND	: ,, [-]

IO(I)	:	,, [-]
J	:	index van de vraagpunten [-]
JH,JH1,JH2	:	hulpindices [-]
K	:	index van de spline functie intervallen [-]
KQ(I,K)	:	zandproduktie van winput I waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KS	:	hulpvariabele [-]
KU	:	aantal spline functie intervallen [-]
LARGE	:	,, [f/m^3]
M	:	aantal vraagpunten [-]
MIN	:	hulpvariabele [f/m^3]
N	:	aantal winputten [-]
NC	:	aantal beperkingen van het gereduceerde primale probleem [-]
NCP	:	aantal beperkingen van het volledige primale probleem [-]
NTP	:	totaal aantal variabelen van het volledige primale probleem (inclusief de spelingsvariabelen) [-]
NV	:	aantal continue variabelen van het gereduceerde primale probleem [-]
NVP	:	aantal continue variabelen van het volledige primale probleem [-]
PRIOPT	:	minimum van het primale probleem [f]
PSI(I,J)	:	hulpvariabele [f/m^3]
RES	:	bijdrage van de dummievariabelen aan PRIOPT [f]
RQ0(I)	:	minimaal te produceren hoeveelheid zand ingeval winput I in produktie genomen wordt [m^3]
S	:	index van de leidingtrajecten [-]
SI(I)	:	aantal trajecten van het leidingennet van winput I [-]
SU	:	aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
SUM	:	hulpvariabele [f/m^3]
SUM1	:	,, [m^3]
SUM2	:	,, [m^3]
X(J)	:	continue variabele [m^3]
X1(J)	:	dummievariabele [-]
XB(J)	:	variabele die aangeeft welke variabelen basisvariabele zijn [-]
XR(J)	:	continue variabele, berekend in de subroutine RESIM [m^3]
ZR	:	bijdrage van de continue variabelen aan PRIOPT [f]

Uitvoerbestand van de subroutinetest

22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41

Gebruikte parameters:

KU : 5 M : 5 N : 2 SU : 86

NCP: 7 NTP: 37 NVP: 17

Tussenuitkomsten:

```
I        : 1     KS   : 0

I        : 2     KS   : 2

NIO      : 2     IO(1): 1     IO(2): 2

BL(I)    : 1000000.0    40000.0    40000.0    40000.0    40000.0
             50000.0   788107.1

I        : 1     SUM:  0.000

I        : 2     SUM: -0.597

ETA(I,J): 1 1 2 2 1
          2 0 0 0 0

PSI(I,J):  6.966    7.666    2.029    2.398    4.516
           2.285   99.999   99.999   99.999   99.999

C(J)     :  6.966    7.666    2.029    2.398    4.516
            2.285   99.999   99.999   99.999   99.999
            0.000    0.000    0.000    0.000    0.000
            0.000    0.000

NC       :  7     NV: 17

AM(I,J)  :  1  0  0  0  0  1  0  0  0  0 -1  0  0  0  0  0  0
            0  1  0  0  0  0  1  0  0  0  0 -1  0  0  0  0  0
            0  0  1  0  0  0  0  1  0  0  0  0 -1  0  0  0  0
            0  0  0  1  0  0  0  0  1  0  0  0  0 -1  0  0  0
            0  0  0  0  1  0  0  0  0  1  0  0  0  0 -1  0  0
            1  1  1  1  1  0  0  0  0  0  0  0  0  0  0 -1  0
            0  0  0  0  0  1  1  1  1  1  0  0  0  0  0  0 -1

II(1)    : 2

II(2)    : 1

II(3)    : 1

II(4)    : 1

II(5)    : 1

X(17)    : 1000000.0

X(16)    :   40000.0

X(16)    :   80000.0

X(16)    :  120000.0

X(15)    :       0.0     X(17): 1000000.0
```

```
SUM1    :   40000.0     SUM2 : -211892.9

IND     : 1

XB(I)   :  6  2  3  4  10 16 17

BI(I,J) :  1.0  0.0  0.0  0.0  0.0  0.0  0.0
           0.0  1.0  0.0  0.0  0.0  0.0  0.0
           0.0  0.0  1.0  0.0  0.0  0.0  0.0
           0.0  0.0  0.0  1.0  0.0  0.0  0.0
           0.0  0.0  0.0  0.0  1.0  0.0  0.0
           0.0  1.0  1.0  1.0  0.0 -1.0  0.0
           1.0  0.0  0.0  0.0  1.0  0.0 -1.0

IND     : 1

BI(I,J) :  1.0  0.0  0.0  0.0  0.0  0.0  0.0
           0.0  1.0  0.0  0.0  0.0  0.0  0.0
           0.0  0.0  1.0  0.0  0.0  0.0  0.0
           0.0  0.0  0.0  1.0  0.0  0.0  0.0
           0.0  0.0  0.0  0.0  1.0  0.0  0.0
           0.0  1.0  1.0  1.0  0.0 -1.0  0.0
           1.0  0.0  0.0  0.0  1.0  0.0 -1.0
XB(I)   :  6  2  3  4  5 16 17

XR(J)   :        0.0     40000.0     40000.0     40000.0     40000.0
       1000000.0          0.0         0.0         0.0         0.0
             0.0          0.0         0.0         0.0         0.0
        110000.0     211892.0
```

Berekende variabelen:

```
ZR    : 2949491.0

X(J)  :        0.0     40000.0         0.0         0.0     40000.0
               0.0         0.0         0.0         0.0         0.0
               0.0         0.0     40000.0     40000.0         0.0
         1000000.0         0.0         0.0         0.0         0.0
               0.0         0.0         0.0         0.0         0.0
          821879.4    211892.0         0.0         0.0         0.0
               0.0         0.0         0.0         0.0         0.0
          110000.0    211892.0

RES   :  756029.4

PRIOPT: 3705520.5
```

Controle d.m.v. handberekening

D4(1,K)=0: I=1: KS=0

D4(2,1)=1, D4(2,2)=1, D4(2,3-5)=0: KS=2

D1(1), D1(2)=1: NIO=2, IO(1)=1, IO(2)=2

BL(1-5)=A(1-5)

I=1: KS=0: BL(6)=RQ0(1)=50000.0

I=2: KS=2: BL(7)=KQ(2,2)=788107.1

D4(1,K)=0: I=1: SUM=0

I=2: SUM=1*-0.533+1*-0.064=-0.597

D1(1)=1, F(1,1)>G(1,1), COEF(1,1,1)=1, D3(1,1)=0:
ETA(1,1)=1 en PSI(1,1)=1*(6.966+0.0)+(1-1)*99.999=6.966

D1(2)=1, F(2,1)>G(2,1), D3(2,S)=COEF(2,S,1):
ETA(2,1)=2, PSI(2,1)=2.883-0.597=2.286

C(1)=PSI(1,1)=6.966, C(6)=PSI(2,1)=2.286

NC=2+5=7, NV=2*5+7=17

AM(I,J): gecontroleerd

J=1, I=1, 6.966<99.999: MIN=6.966, II(1)=1
I=2, 2.885< 6.966: MIN=2.885, II(1)=2

J=1: XB(1)=(2-1)*5+1=6, JH=2*5+5+2=17, X(17)=1000000.0

J=2: XB(2)=(1-1)*5+2=2, JH=2*5+5+1=16, X(16)=40000.0

J=3: XB(3)=(1-1)*5+3=3, JH=2*5+5+1=16, X(16)=40000.0+40000.0=80000.0

J=4: XB(4)=(1-1)*5+4=4, JH=2*5+5+1=16, X(16)=40000.0+40000.0+40000.0
=120000.0

J=6, JH=2*5+6=16, 120000.0>50000.0: XB(6)=16

SUM1=40000.0-0.0=40000.0, SUM2=788107.1-1000000.0=-211892.9

XB(5)=2*5=10, XB(7)=2*5+5=15

-211892.9≤40000.0: IND=1, XB(7)=17

Controle BI(I,J): BI*(I,J)*BI(I,J)=I

met BI*(I,J): de coëfficiëntenmatrix van de basisvariabelen voor het inverteren

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

De uitkomsten van RESIM zijn niet gecontroleerd; de subroutine RESIM is in bijlage XXI al gecontroleerd.

```
RES=148435.8*1+15000*1+39465.0*1+26310.0*1+48235.0*1
    +151989.5*1+67920.0*1+50940.0*1+62260.0*1
    -(-0.533*178120.6-0.064*788107.1)=755931.9≈756029.4

J=1, JH1=2*2*5+2*5+1=31, JH2=2*5+1=11: X(31)=X(11)=0.0, X(11)=0.0

I=1, IH1=2*2*5+2*5+5+1=36, IH2=2*5+5+1=16: X(36)=X(16)=110000.0,
                                           X(16)=0.0

I=2, J=1, JH1=2*5+(2-1)*5+1=16, JH2=(2-1)*5+1=6: X(16)=X(6)=1000000.0
                                                 X(6)=0.0

I=2: SUM=1000000.0

I=2, K=1, J=2*2*5+(2-1)*5+1=26: X(26)=1000000.0-178120.6=821879.4
```

Listing van het programma TESTDUAL

```
      PROGRAM TESTDUAL
C
C     Declareren van de parameters
C
      INTEGER M,N
C
C     Definiëren van de parameters
C
      PARAMETER (M=5,N=3)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER I2,IU2(N),J2,JU2(M)
      REAL AB,U0(M),Z(N)
C
C     Declareren van de lokale variabelen
C
      INTEGER GI(N),IND,J1(M),NJ1,NJ2,NNV(M),NV(N,M),NVL,PSI(N,M)
C
C     Declareren van de indices
C
      INTEGER I,J,JH,JH1,JH2
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*57 PROBLEM
C
C     Communiceren met de subroutine CORRECT
C
      COMMON/BLOCK1/AB,I2,IU2,J2,JU2,U0,Z
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) PROBLEM
      WRITE(2,25) DATE,PROBLEM
C
C     Inlezen en uitvoeren van de invoergegevens
C
      READ(1,*) GI(1),(NNV(J),J=1,M)
      DO 20 J=1,M
            DO 10 JH=1,NNV(J)
                  READ(1,*) NV(J,JH)
   10       CONTINUE
   20 CONTINUE
      DO 40 J=1,M
            DO 30 JH=1,NNV(J)
                  I=NV(J,JH)
                  READ(1,*) PSI(I,J)
   30       CONTINUE
```

```
   40 CONTINUE

      WRITE(2,35)
      WRITE(2,45) GI(1)
      WRITE(2,55) (J,NNV(J),J=1,M)
      WRITE(2,65) ((J,NNV(J),NV(J,JH),JH=1,NNV(J)),J=1,M)
      WRITE(2,75) ((NV(J,JH),J,PSI(NV(J,JH),J),JH=1,NNV(J)),J=1,M)
C
C     Stap 1, inclusief het uitvoeren van tussenuitkomsten
C
      Z(1)=0.0
      DO 50 I=2,N
            Z(I)=0.0
            GI(I)=0.0
   50 CONTINUE
      WRITE(2,85)
      WRITE(2,95) (I,GI(I),I,Z(I),I=1,N)
C
C     Stap 2, inclusief het uitvoeren van tussenuitkomsten
C
      NJ1=0
      DO 90 J=1,M
            U0(J)=0.0
            DO 60 JH=1,NNV(J)
                  I=NV(J,JH)
                  IF(GI(I).EQ.1)GOTO 70
   60       CONTINUE
            NJ1=NJ1+1
            J1(NJ1)=J
            GOTO 90
   70       CONTINUE
            U0(J)=PSI(I,J)-Z(I)
            DO 80 JH=1,NNV(J)
                  I=NV(J,JH)
                  Z(I)=PSI(I,J)-U0(J)
                  GI(I)=1
   80       CONTINUE
   90 CONTINUE
      WRITE(2,105)
      WRITE(2,95) (I,GI(I),I,Z(I),I=1,N)
      WRITE(2,115) (J,U0(J),J=1,M)
      IF(NJ1.NE.0)THEN
        WRITE(2,125) NJ1,(J,J1(J),J=1,NJ1)
      ELSE
        WRITE(2,125) NJ1
      ENDIF
C
C     Stap 3, inclusief het uitvoeren van tussenuitkomsten
C
      IF(NJ1.EQ.0)GOTO 240
      NVL=0
  100 CONTINUE
      NJ2=0
      IND=0
      DO 140 JH1=1,NJ1
            J=J1(JH1)
```

```
            DO 110 JH2=1,NNV(J)
                   IF(NV(J,J2).EQ.1)GOTO 120
110         CONTINUE
            NJ2=NJ2+1
            J1(NJ2)=J
            NVL=NVL+NNV(J)-1
            GOTO 140
120         CONTINUE
            U0(J)=PSI(I,J)-Z(I)
            IND=1
            DO 130 JH2=1,NNV(J)
                   I=NV(J,JH2)
                   Z(I)=PSI(I,J)-U0(J)
                   GI(I)=1
130         CONTINUE
140 CONTINUE
    WRITE(2,135)
    WRITE(2,95) (I,GI(I),I,Z(I),I=1,N)
    WRITE(2,115) (J,U0(J),J=1,M)
    IF(NJ2.NE.0)THEN
      WRITE(2,145) NJ2,(J,J1(J),J=1,NJ2)
    ELSE
      WRITE(2,145) NJ2
    ENDIF
    WRITE(2,155) NVL,IND
C
C   Stap 4, inclusief het uitvoeren van tussenuitkomsten
C
    IF(NJ2.EQ.0)GOTO 240
    NJ1=NJ2
    IF(IND.EQ.1)GOTO 100
    IF(NVL.EQ.0)THEN
      DO 150 JH=1,NJ1
             J=J1(JH)
             I=NV(J,1)
             U0(J)=PSI(I,J)
             GI(I)=1
150   CONTINUE
      GOTO 240
    ENDIF
    WRITE(2,165)
    WRITE(2,175) (I,GI(I),I=1,N)
    WRITE(2,115) (J,U0(J),J=1,M)
    WRITE(2,125) NJ1
C
C   Stap 5, inclusief het uitvoeren van tussenuitkomsten
C
160 CONTINE
    J=J1(NJ1)
    I=NV(J,1)
    U0(J)=PSI(I,J)
    NJ1=NJ1-1
    NVL=NVL-NNV(J)+1
    I2=0
    J2=1
    JU2(J2)=J
```

```
      DO 170 JH=1,NNV(J)
             I=NV(J,JH)
             Z(I)=PSI(I,J)-U0(J)
             I2=I2+1-GI(I)
             IF(GI(I).EQ.0)IU2(I2)=I
             GI(I)=1
             AB=Z(I)
             IF(AB.LT.0.0)CALL CORRECT
  170 CONTINUE
      WRITE(2,185)
      WRITE(2,95) (I,GI(I),I,ZI(I),I=1,N)
      WRITE(2,115) (J,U0(J),J=1,M)
      WRITE(2,125) NJ1
      WRITE(2,145) NVL
      WRITE(2,195) I2,(I,IU2(I),I=1,I2)
      WRITE(2,205) J2,(I,JU2(I),J=1,J2)
      WRITE(2,215) AB
C
C     Stap 6, inclusief het uitvoeren van tussenuitkomsten
C
      IF(NJ1.EQ.0)GOTO 240
  180 CONTINUE
      NJ2=0
      IND=0
      DO 220 JH1=1,NJ1
             J=J1(JH1)
             DO 190 JH2=1,NNV(J)
                    I=NV(J,JH2)
                    IF(GI(I).EQ.1)GOTO 200
  190        CONTINUE
             NJ2=NJ2+1
             J1(NJ2)=J
             GOTO 220
  200        CONTINUE
             U0(J)=PSI(I,J)-Z(I)
             IND=1
             NVL=NVL-NNV(J)+1
             J2=J2+1
             JU2(J2)=J
             AB=-U0(J)
             IF(AB.GT.0.0)CALL CORRECT
             DO 210 JH2=1,NNV(J)
                    I=NV(J,JH2)
                    Z(I)=PSI(I,J)-U0(J)
                    I2=I2+1-GI(I)
                    IF(GI(I).EQ.0)IU2(I2)=I
                    GI(I)=1
                    AB=Z(I)
                    IF(AB.LT.0.0)CALL CORRECT
  210        CONTINUE
  220 CONTINUE
      WRITE(2,225)
      WRITE(2,95) (I,GI(I),I,ZI(I),I=1,N)
      WRITE(2,145) NJ2
      WRITE(2,155) NVL,IND
      WRITE(2,195) I2,(I,IU2(I),I=1,I2)
```

```
      WRITE(2,205) J2,(J,JU2(I),J=1,J2)
      WRITE(2,215) AB
      IF(NJ2.EQ.0)GOTO 240
      NJ1=NJ2
      IF(IND.EQ.1)GOTO 180
      IF(NVL.GT.0)GOTO 160
      DO 230 JH=1,NJ2
             J=J1(JH)
             I=NV(J,1)
             U0(J)=PSI(I,J)-Z(I)
             GI(I)=1
  230 CONTINUE
      WRITE(2,175) (I,GI(I),I=1,N)
      WRITE(2,95) (J,U0(J),J=1,M)
C
C     Uitvoeren van de berekende variabelen
C
  240 CONTINUE
      WRITE(2,235)
      WRITE(2,245) (I,ZI(I),I=1,N)
      WRITE(2,115) (J,U0(J),J=1,M)
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A57)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,'Ingelezen gegevens:')
   45 FORMAT(/X,'GI(1)',3X,':',X,I1)
   55 FORMAT(/X,'NNV(',I1,')',2X,':',X,I1,4(4X,'NNV(',I1,')',2X,
             ':',X,I1))
   65 FORMAT(/(X,'NV(',I1,',',I1,')',X,':',I1,2(4X,'NV(',I1,',',I1,
     #       ')',X,':',I1)))
   75 FORMAT(/(X,'PSI(',I1,',',I1,'):',X,F2.1,2(2X,'PSI(',I1,',',
     #       I1,'):',X,F2.1)))
   85 FORMAT(//X,'Tussenuitkomsten:'//X,'Stap 1:')
   95 FORMAT(/3(X,'GI(',I1,'):',X,I1,5X,'Z(',I1,')',2X,':',X,I1))
  105 FORMAT(//X,'Stap 2:')
  115 FORMAT(/X,'U0(',I1,'):',X,F2.1,4(3X,'U0(',I1,'):',X,F2.1))
  125 FORMAT(/X,'NJ1',2X,':',X,I1,2(5X,'J1(',I1,')',X,':',X,I1))
  135 FORMAT(//X,'Stap 3:')
  145 FORMAT((/X,'NJ2',2X,':',X,I1,2(5X,'J1(',I1,')',X,':',X,I1))
  155 FORMAT(/X,'NVL',2X,':',X,I1,5X,'IND',3X,':',I1)
  165 FORMAT(//X,'Stap 4:')
  175 FORMAT(/X,'GI(',I1,'):',X,I1,2(5X,'GI(',I1,')',X,':',X,I1))
  185 FORMAT(//X,'Stap 5:')
  195 FORMAT(/X,'I2',3X,':',X,I1,2(5X,'IU(',I1,'):',X,I1))
  205 FORMAT(/X,'J2',3X,':',X,I1,2(5X,'JU(',I1,'):',X,I1))
  215 FORMAT(/X,'AB',3X,':',X,F3.1)
  225 FORMAT((//X,'Stap 6:')
  235 FORMAT(//X,'Berekende variabelen:')
  245 FORMAT(/X,'Z(',I1,')',X,':',X,F2.1,2(3X,'Z(',I1,')',X,':',
             X,F2.1))
      END
```

Listing van de subroutine CORRECT

```
      SUBROUTINE CORRECT
C
C     Declareren van de parameters
C
      INTEGER M,N
C
C     Definiëren van de parameters
C
      PARAMETER (M=5,N=3)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER I2,IU2(N),J2,JU2(M)
      REAL AB,U0(M),Z(N)
C
C     Declareren van de indices
C
      INTEGER I,IH,J,JH
C
C     Communiceren met de subroutine TESTDUAL
C
      COMMON/BLOCK1/AB,I2,IU2,J2,JU2,U0,Z
C
      DO 10 IH=1,I2
            I=IU2(IH)
            Z(I)=Z(I)-AB
   10 CONTINUE
      DO 20 JH=1,J2
            J=JU2(JH)
            U0(J)=U0(J)+AB
   20 CONTINUE
      RETURN
      END
```

Betekenis van de in het programma TESTDUAL en de subroutine CORRECT gebruikte variabelen

AB	: waarde van de duale variabele [-]
DATE	: datum van de programma-run [-]
GI(I)	: 0-1 variabele die aangeeft of Z(I) gewaardeerd is [-]
I	: index van variabele Z(I) [-]
I2	: hulpindex [-]
IH	: hulpindex [-]
IND	: 0-1 variabele die aangeeft of een U0(J) in de huidige iteratiestap van TESTDUAL gewaardeerd is [-]
IU2(I)	: hulpvariabele [-]
J	: index van de variabele U0(J) [-]
J1(J)	: verzameling indices J van de variabelen U0(J) die nog niet gewaardeerd zijn [-]
J2	: hulpindex [-]
JH,JH1,JH2	: hulpindices [-]
JU2(J)	: hulpvariabele [-]

M	:	aantal U0(J) [-]
N	:	aantal Z(I) [-]
NJ1	:	aantal indices J1(J) [-]
NJ2	:	aantal indices J1(J) [-]
NNV(J)	:	aantal I behorende bij een bepaalde J [-]
NV(J,I)	:	indices I behorende bij een bepaalde J [-]
NVL	:	aantal variabelen U0(J) dat nog niet gewaardeerd is minus het aantal vergelijkingen dat nog niet gebruikt is om een variabele U0(J) of Z(I) te waarderen [-]
PROBLEM	:	identificatie van het testprobleem [-]
PSI(I,J)	:	hulpvariabele [-]
U0(J)	:	duale variabele [-]
Z(I)	:	duale variabele [-]
AB	:	waarde van de duale variabele [-]
I	:	index van de winputten [-]

Invoerbestand van het testprobleem

```
22-10-1986
Invoer: stelsel vergelijkingen in U0(J), Z(I) en PSI(I,J)
1,1,1,1,2,1,
1,
1,
1,
2,
3,
2,
1.0,
1.0,
1.0,
2.0,
1.0,
1.0
```

Uitvoerbestand van het testprobleem

```
22-10-1986
Invoer: stelsel vergelijkingen in U0(J), Z(I) en PSI(I,J)


Ingelezen gegevens:

GI(1)    : 1

NNV(1)   : 1    NNV(2)   : 1    NNV(3)   : 1    NNV(4)   : 2    NNV(5)   : 1

NV(1,1)  : 1    NV(2,1)  : 1    NV(3,1)  : 1

NV(4,1)  : 2    NV(4,2)  : 3    NV(5,1)  : 2

PSI(1,1): 1.0   PSI(1,2): 1.0   PSI(1,3): 1.0

PSI(2,4): 2.0   PSI(3,4): 1.0   PSI(2,5): 1.0
```

Tussenuitkomsten:

Stap 1:

```
GI(1): 1      Z(1)   : 0.0
GI(2): 0      Z(2)   : 0.0
GI(3): 0      Z(3)   : 0.0
```

Stap 2:

```
GI(1): 1      Z(1)   : 0.0
GI(2): 0      Z(2)   : 0.0
GI(3): 0      Z(3)   : 0.0

U0(1): 1.0    U0(2) : 1.0    U0(3) : 1.0    U0(4) : 0.0    U0(5) : 0.0

NJ1   : 2     J1(1) : 4      J1(2) : 5
```

Stap 3:

```
GI(1): 1      Z(1)   : 0.0
GI(2): 0      Z(2)   : 0.0
GI(3): 0      Z(3)   : 0.0

U0(1): 1.0    U0(2) : 1.0    U0(3) : 1.0    U0(4) : 0.0    U0(5) : 0.0

NJ2   : 2     J1(1) : 4      J1(2) : 5

NVL   : 1     IND    : 0
```

Stap 4:

```
GI(1): 1      GI(2) : 0      GI(3) : 0

U0(1): 1.0    U0(2) : 1.0    U0(3) : 1.0    U0(4) : 0.0    U0(5) : 0.0

NJ1   : 2
```

Stap 5:

```
GI(1): 1      Z(1)   : 0.0
GI(2): 1      Z(2)   : 0.0
GI(3): 0      Z(3)   : 0.0

U0(1): 1.0    U0(2) : 1.0    U0(3) : 1.0    U0(4) : 0.0    U0(5) : 1.0

NJ1   : 1

NVL   : 1

I2    : 1     IU2(1): 1
```

```
J2   : 1     JU2(1): 5

AB   :  0.0
```

Stap 6:

```
GI(1): 1     Z(1)  : 0.0
GI(2): 1     Z(2)  : 1.0
GI(3): 1     Z(3)  : 0.0

NJ2  : 0

NVL  : 0     IND   : 1

I2   : 2     IU2(1): 2     IU2(2): 3
J2   : 2     JU2(1): 5     JU2(2): 4
AB   : -1.0
```

Berekende variabelen:

```
Z(1) : 0.0   Z(2)  : 1.0   Z(3)  : 0.0

U0(1): 1.0   U0(2) : 1.0   U0(3) : 1.0   U0(4) : 1.0   U0(5) : 0.0
```

BIJLAGE XXIV

Listing van het programma TESTTRANS

```
      PROGRAM TESTTRANS
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,ND,NTP,SU,TU
C
C     Definiëren van de parameters
C
      PARAMETER (KU=5,M=5,N=2,SU=25,TU=1)
      PARAMETER (ND=2*N+N*SU+N*KU,NTP=2*N*M+N+M+N*KU)
C
C     Declareren van de niet-lokale variabelen
C
      REAL BE(TU),DUOPT,GAM(TU,ND)
C
C     Declareren van de indices
C
      INTEGER J,T
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*62 AREA
C
C     Communiceren met de subroutine TRANS
C
      COMMON/BLOCK1/ BE,DUOPT,GAM
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) AREA
      WRITE(2,25) DATE,AREA
C
C     Uitvoeren van de gebruikte parameters
C
      WRITE(2,35)
      WRITE(2,45) KU,M,N,SU,TU,ND,NTP
C
C     Aanroepen van de subroutines RECALCO, ANSOL en PRISOL
C
      CALL RECALCO
      CALL ANSOL
      CALL PRISOL(*10,*20)
   10 CONTINUE
      T=1
C
C     Aanroepen van de subroutine TRANS
C
      CALL TRANS(T)
```

```
C
C     Uitvoeren van de berekende variabelen
C
      WRITE(2,55)
      WRITE(2,65) T,BE(T)
      WRITE(2,75) T,(GAM(T,J),J=1,ND)
      WRITE(2,85) DUOPT
   20 CONTINUE
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A62)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,'Gebruikte parameters:')
   45 FORMAT(/X,'KU:',X,I2,5X,'M',2X,':',X,I2,5X,'N:',X,I1,5X,'SU:',
     #       X,I2,5X,'TU:',X,I1/X,'ND:',X,I2,5X,'NTP:',X,I2)
   55 FORMAT(//X,'Berekende variabelen:')
   65 FORMAT(/X,'BE(',I1,')',3X,':',X,F9.1)
   75 FORMAT(/X,'GAM(',I1,',J):',X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
   85 FORMAT(/X,'DUOPT',3X,':',X,F9.1)
      END
```

Listing van de subroutine RECALCO

De listing van de subroutine RECALCO is identiek aan die van bijlage XIX met uitzondering van de volgende COMMON-statements:

```
C
C     Communiceren met de subroutines ANSOL, PRISOL en TRANS
C
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK3/ Q,QA,QBU
```

Listing van de subroutine ANSOL

De listing van de subroutine ANSOL is identiek aan die van bijlage XX zonder het uitvoeren van de tussenuitkomsten en met uitzondering van de volgende COMMON-statements:

```
C
C     Communiceren met de subroutines RECALCO, PRISOL en TRANS
C
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK3/ Q,QA,QBU
      COMMON/BLOCK4/ D1,D2,D3,D4
```

Listing van de subroutine PRISOL

De listing van de subroutine PRISOL is identiek aan die van bijlage XXII zonder het uitvoeren van de tussenuitkomsten en met uitzondering van de volgende COMMON-statements:

```
C
C     Communiceren met de subroutines RECALCO, ANSOL, RESIM en TRANS
C
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK4/ D1,D2,D3,D4
      COMMON/BLOCK5/ PSI,X
      COMMON/BLOCK6/ AM,BL,BI,C,NC,NV,XB,XR
```

Listing van de subroutine RESIM

De listing van de subroutine RESIM is identiek aan die van bijlage XXI met uitzondering van de volgende PARAMETER- en COMMON-statements:

```
      PARAMETER (M=5,N=2)
      PARAMETER (NCP=N+M,NVP=N*M+N+M)
C
C     Communiceren met de subroutine PRISOL
C
      COMMON/BLOCK6/ A,B,BI,C,NC,NV,XB,X
```

Listing van de subroutine TRANS

```
      SUBROUTINE TRANS(T)
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,ND,NTP,SU,TU
C
C     Definiëren van de parameters
C
      PARAMETER (KU=5,M=5,N=2,SU=25,TU=1)
      PARAMETER (ND=2*N+N*SU+N*KU,NTP=2*N*M+N+M+N*KU)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),B(N),COEF(N,SU,M),D1(N),D2(N),D3(N,SU),D4(N,KU),
     #        I2,IU2(N),J2,JU2(M),QA,QBU(N,SU),
      REAL AB,AU(N),BE(TU),DUOPT,E(N,SU),F(N,M),G(N,M),GAM(TU,ND),
     #     H(N,KU),KQ(N,KU),PSI(N,M),Q(N,KU),RQ0(N),U0(M),X(NTP-M-N),
     #     YS0(M),YS1(N),Z(N)
C
C     Declareren van de lokale variabelen
C
      INTEGER GI(N),IND,J1(M),KS,NJ1,NJ2,NNV(M),NV(M,N),NVL
      REAL HELP1,HELP2,QX,SUM,U4(N,KU),U5(N,KU)
C
C     Declareren van de indices
C
      INTEGER I,IH1,IH2,J,JH,JH1,JH2,K,S,T
C
C     Communiceren met het programma TESTTRANS en de subroutines
C     RECALCO, ANSOL, PRISOL en CORRECT
C
```

```
      COMMON/BLOCK1/ BE,GAM,DUOPT
      COMMON/BLOCK2/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK3/ Q,QA,QBU
      COMMON/BLOCK4/ D1,D2,D3,D4
      COMMON/BLOCK5/ PSI,X,YS0,YS1
      COMMON/BLOCK7/ AB,I2,IU2,J2,JU2,U0,Z
C
C     Bepalen van NNV(J), NV(J,NNV(J)) en uitvoeren van tussen-
C     uitkomsten
C
      DO 20 J=1,M
            NNV(J)=0
            DO 10 I=1,N
                  IH1=(I-1)*M+J
                  IH2=N*M+IH1
                  IF((X(IH1)+X(IH2)).EQ.0.0)GOTO 10
                  NNV(J)=NNV(J)+1
                  NV(J,NNV(J))=I
   10       CONTINUE
   20 CONTINUE
      WRITE(2,5)
      WRITE(2,15) (J,NNV(J),J=1,M)
      WRITE(2,25) ((J,JH,NV(J,JH),JH=1,NNV(J)),J=1,M)
C
C     Stap 1, inclusief het uitvoeren van tussenuitkomsten
C
      WRITE(2,35)
      DO 40 I=1,N
            Z(I)=0.0
            GI(I)=1
            IF(D1(I).EQ.0)GOTO 40
            KS=0
            DO 30 K=1,KU
                  KS=KS+D4(I,K)
   30       CONTINUE
            QX=X(2*N*M+(I-1)*KU+KS)
            WRITE(2,45) I,QX,YS1(I)
            IF(((1-D4(I,1))*YS1(I)+D4(I,1)*QX).GT.0.0)GOTO 40
            GI(I)=0
   40 CONTINUE
      WRITE(2,55) (I,GI(I),I,Z(I),I=1,N)
C
C     Stap 2, inclusief het uitvoeren van tussenuitkomsten
C
      WRITE(2,65)
      NJ1=0
      DO 90 J=1,M
            U0(J)=0.0
            WRITE(2,75) J,YS0(J)
            IF(YS0(J).GT.0.0)GOTO 70
            DO 50 JH=1,NNV(J)
                  I=NV(J,JH)
                  IF(GI(I).EQ.1)GOTO 60
   50       CONTINUE
            NJ1=NJ1+1
            J1(NJ1)=J
```

```
            GOTO 90
   60       CONTINUE
            U0(J)=PSI(I,J)-Z(I)
   70       CONTINUE
            DO 80 JH=1,NNV(J)
                  I=NV(J,JH)
                  Z(I)=PSI(I,J)-U0(J)
                  GI(I)=1
   80       CONTINUE
   90 CONTINUE
      WRITE(2,55) (I,GI(I),I,Z(I),I=1,N)
      WRITE(2,85) (J,UO(J),J=1,M)
      IF(NJ1.NE.0)THEN
        WRITE(2,95) NJ1,(J,J1(J),J=1,NJ1)
      ELSE
        WRITE(2,95) NJ1
      ENDIF
C
C     Stap 3, inclusief herhaling stap 2 en het uitvoeren van tussen-
C     uitkomsten
C
      IF(NJ1.EQ.0)GOTO 240
      NVL=0
  100 CONTINUE
      NJ2=0
      IND=0
      DO 140 JH1=1,NJ1
             J=J1(JH1)
             DO 110 JH2=1,NNV(J)
                    I=NV(J,JH2)
                    IF(GI(I).EQ.1)GOTO 120
  110        CONTINUE
             NJ2=NJ2+1
             J1(NJ2)=J
             NVL=NVL+NNV(J)-1
             GOTO 140
  120        CONTINUE
             U0(J)=PSI(I,J)-Z(I)
             IND=1
             DO 130 JH2=1,NNV(J)
                    I=NV(J,JH2)
                    Z(I)=PSI(I,J)-U0(J)
                    G(I)=1
  130        CONTINUE
  140 CONTINUE
      WRITE(2,105)
      WRITE(2,55) (I,GI(I),I,Z(I),I=1,N)
      WRITE(2,85) (J,UO(J),J=1,M)
      IF(NJ2.NE.0)THEN
        WRITE(2,115) NJ2,(J,J1(J),J=1,NJ2)
      ELSE
        WRITE(2,115) NJ2
      ENDIF
      WRITE(2,125) IND,NVL
```

```
C
C     Stap 4, inclusief het uitvoeren van tussenuitkomsten
C
      IF(NJ2.EQ.0)GOTO 240
      NJ1=NJ2
      IF(IND.EQ.1)GOTO 100
      IF(NVL.EQ.0)THEN
        DO 150 JH=1,NJ1
               J=J1(JH)
               I=NV(J,1)
               U0(J)=PSI(I,J)
               GI(I)=1
  150   CONTINUE
        GOTO 240
      ENDIF
      WRITE(2,135)
      WRITE(2,145) (I,GI(I),I=1,N)
      WRITE(2,85) (J,UO(J),J=1,M)
      WRITE(2,95) NJ1
C
C     Stap 5, inclusief herhaling stap 2 en het uitvoeren van tussen-
C     uitkomsten
C
  160 CONTINUE
      J=J1(NJ1)
      I=NV(J,1)
      U0(J)=PSI(I,J)
      NJ1=NJ1-1
      NVL=NVL-NNV(J)+1
      I2=0
      J2=1
      JU2(J2)=J
      DO 170 JH=1,NNV(J)
             I=NV(J,JH)
             Z(I)=PSI(I,J)-U0(J)
             I2=I2+1-GI(I)
             IF(GI(I).EQ.0)IU2(I2)=I
             GI(I)=1
             AB=Z(I)
             IF(AB.LT.0.0)CALL CORRECT
  170 CONTINUE
      WRITE(2,155)
      WRITE(2,55) (I,GI(I),I,Z(I),I=1,N)
      WRITE(2,85) (J,UO(J),J=1,M)
      WRITE(2,95) NJ1
      WRITE(2,165) NVL
      WRITE(2,175) I2,(I,IU2(I),I=1,I2)
      WRITE(2,185) J2,(J,JU2(J),J=1,J2)
      WRITE(2,195) AB
      IF(NJ1.EQ.0)GOTO 240
  180 CONTINUE
      NJ2=0
      IND=0
      DO 220 JH1=1,NJ1
             J=J1(JH1)
             DO 190 JH2=1,NNV(J)
```

```
                I=NV(J,JH2)
                IF(GI(I).EQ.1)GOTO 200
190         CONTINUE
            NJ2=NJ2+1
            J1(NJ2)=J
            GOTO 220
200         CONTINUE
            U0(J)=PSI(I,J)-Z(I)
            IND=1
            NVL=NVL-NNV(J)+1
            J2=J2+1
            JU2(J2)=J
            AB=-U0(J)
            IF(AB.GT.0.0)CALL CORRECT
            DO 210 JH2=1,NNV(J)
                   I=NV(J,JH2)
                   Z(I)=PSI(I,J)-U0(J)
                   I2=I2+1-GI(I)
                   IF(GI(I).EQ.0)IU2(I2)=I
                   GI(I)=1
                   AB=Z(I)
                   IF(AB.LT.0.0)CALL CORRECT
210         CONTINUE
220 CONTINUE
    WRITE(2,55) (I,GI(I),I,Z(I),I=1,N)
    WRITE(2,115) NJ2
    WRITE(2,125) IND,NVL
    WRITE(2,175) I2,(I,IU2(I),I=1,I2)
    WRITE(2,185) J2,(J,JU2(J),J=1,J2)
    IF(NJ2.EQ.0)GOTO 240
    NJ1=NJ2
    IF(IND.EQ.1)GOTO 180
    IF(NVL.GT.0)GOTO 160
    DO 230 JH=1,NJ2
           J=J1(JH)
           I=NV(J,1)
           U0(J)=PSI(I,J)-Z(I)
           GI(I)=1
230 CONTINUE
    WRITE(2,145) (I,GI(I),I=1,N)
    WRITE(2,85) (J,UO(J),J=1,M)
C
C     Stap 6, inclusief het uitvoeren van tussenuitkomsten
C
240 CONTINUE
    WRITE(2,205)
    WRITE(2,215) (I,Z(I),I=1,N)
    WRITE(2,85) (J,UO(J),J=1,M)
    BE(T)=0.0
    DO 250 J=1,M
           BE(T)=BE(T)+U0(J)*A(J)
250 CONTINUE
    DUOPT=BE(T)
    DO 280 I=1,N
           KS=0
           DO 260 K=1,KU
```

```
                KS=KS+D4(I,K)
                U4(I,K)=0.0
                U5(I,K)=-H(I,K)
260         CONTINUE
            IF(KS.EQ.0)GOTO 280
            DO 270 K=1,KS
                U4(I,K)=-H(I,K)
                U5(I,K)=0.0
287         CONTINUE
            U4(I,KS)=U4(I,KS)+Z(I)
280 CONTINUE
    DO 350 I=1,N
            HELP1=0.0
            IF(D1(I).EQ.0)THEN
              HELP1=U0(1)-F(I,1)
              IF(G(I,1).LT.F(I,1))HELP1=U0(1)-G(I,1)
              DO 290 J=2,M
                    HELP2=U0(J)-F(I,J)
                    IF(G(I,J).LT.F(I,J))HELP2=U0(J)-G(I,J)
                    IF(HELP2.GT.HELP1)HELP1=HELP2
290           CONTINUE
              IF(HELP1.LT.0.0)HELP1=0.0
            ENDIF
            GAM(T,I)=AU(I)+RQ0(I)*(1-D4(I,1))*Z(I)-HELP1*QA
            DUOPT=DUOPT+D1(I)*GAM(T,I)
            HELP1=0.0
            SUM=0.0
            IF(D4(I,1).EQ.1)THEN
              DO 300 K=1,KU
                    SUM=SUM+D4(I,K)*H(I,K)
300           CONTINUE
            ENDIF
            IF(D2(I).EQ.0)THEN
              HELP1=U0(1)-F(I,1)-SUM
              DO 310 J=2,M
                    HELP2=U0(J)-F(I,J)-SUM
                    IF(HELP2.GT.HELP1)HELP1=HELP2
310           CONTINUE
              IF(HELP1.LT.0.0)HELP1=0.0
            ENDIF
            GAM(T,N+I)=B(I)-HELP1*QA
            DUOPT=DUOPT+D2(I)*GAM(T,N+I)
            DO 330 S=1,SU
                    HELP1=0.0
                    IF(D3(I,S).EQ.0)THEN
                      HELP1=COEF(I,S,1)*(U0(1)-G(I,1)-SUM)
                      DO 320 J=2,M
                            HELP2=COEF(I,S,J)*(U0(J)-G(I,J)-SUM)
                            IF(HELP2.GT.HELP1)HELP1=HELP2
320                   CONTINUE
                      IF(HELP1.LT.0.0)HELP1=0.0
                    ENDIF
                    JH=2*N+(S-1)*N+I
                    GAM(T,JH)=E(I,S)-HELP1*QBU(I,S)
                    DUOPT=DUOPT+D3(I,S)*GAM(T,JH)
330         CONTINUE
```

```
            DO 340 K=1,KU
                   JH=2*N+N*SU+(K-1)*N+I
                   GAM(T,JH)=KQ(I,K)*U4(I,K)-Q(I,K)*U5(I,K)
                   DUOPT=DUOPT+D4(I,K)*GAM(T,JH)
  340       CONTINUE
  350 CONTINUE
      RETURN
C
C     Format statements
C
    5 FORMAT(//X,'Tussenuitkomsten:')
   15 FORMAT(/X,'NNV(',I1,')',X,':',X,I1,4(4X,'NNV(',I1,')',X,
            ':',X,I1))
   25 FORMAT(/X,'NV(',I1,',',I1,'):',X,I1,4(4X,'NV(',I1,',',I1,
     #      '):',X,I1))
   35 FORMAT(//X,'Stap 1:')
   45 FORMAT(/X,'I',6X,':',X,I1,4X,'QX:',X,F9.1,6X,'YS1(I):',X,F9.1)
   55 FORMAT(/X,'GI(',I1,')',2X,':',X,I1,4X,'Z(',I1,')',X,':',X,F5.3)
   65 FORMAT(//X,'Stap 2:')
   75 FORMAT(/X,'YSO(',I1,')',X,':',X,F9.1)
   85 FORMAT(/X,'U0(',I1,')',2X,':',X,F5.3)
   95 FORMAT(/X,'NJ1',4X,':',X,I1,4(4X,'J1(',I1,'):',X,I1))
  105 FORMAT(//X,'Stap 3:')
  115 FORMAT(/X,'NJ2',4X,':',X,I1,4(4X,'J1(',I1,'):',X,I1))
  125 FORMAT(/X,'IND',4X,':',X,I1,'NVL',2X,':',X,I1)
  135 FORMAT(//X,'Stap 4:')
  145 FORMAT(/X,'GI(',I1,')',2X,':',X,I1,4X,'GI(',I1,'):',X,I1)
  155 FORMAT(//X,'Stap 5:')
  165 FORMAT(/X,'NVL',4X,':',X,I1)
  175 FORMAT(/X,'I2',5X,':',X,I1,X,4(3X,'IU2(',I1,'):',X,I1))
  185 FORMAT(/X,'J2',5X,':',X,I1,X,4(3X,'JU2(',I1,'):',X,I1))
  195 FORMAT(/X,'AB',5X,':',X,F6.3)
  205 FORMAT(//X,'Stap 6:')
  215 FORMAT(/X,'Z(',I1,')',3X,':',X,F5.3,12X,'Z(',I1,')',2X,':',
     #      X,F5.3)
      END
```

Listing van de subroutine CORRECT

De listing van de subroutine CORRECT is gelijk aan die van bijlage XXIII met uitzondering van de volgende COMMON-statement:

```
C
C     Communiceren met de subroutine TRANS
C
      COMMON/BLOCK7/ AB,I2,IU2,J2,JU2,U0,Z
```

Betekenis van de in de het programma TESTTRANS en de subroutine TRANS gebruikte variabelen

A(J) : vraag naar zand van vraagpunt J [m^3]
AB : waarde van de duale variabele [f/m^3]
AREA : identificatie van het proefgebied [-]

AU(I)	:	parameter voor de vaste kosten van het in produktie nemen van winput I [f]
B(I)	:	parameter voor de vaste kosten van het transport per vrachtwagen vanuit winput I [f]
BE(T)	:	constante van de beperking T in de subroutine ZEON [f]
COEF(I,S,J)	:	0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
DATE	:	datum van de programma-run [-]
DUOPT	:	maximum van het duale probleem [f]
E(I,S)	:	parameter voor de vaste kosten van traject S van het leidingennet van winput I [f]
F(I,J)	:	coëfficiënt voor de variabele kosten van het transport per vrachtwagen van winput I naar vraagpunt J [f/m^3]
G(I,J)	:	coëfficiënt voor de variabele kosten van het transport per pijpleiding van winput I naar vraagpunt J [f/m^3]
GAM(T,J)	:	coëfficiënten van de dummievariabelen van de beperking T in de subroutine ZEON [f]
GI(I)	:	0-1 variabele die aangeeft of Z(I) gewaardeerd is [-]
H(I,J)	:	coëfficiënt voor de reductie in variabele kosten wanneer de produktie van winput I een hoeveelheid KQ(I,K) overschrijdt [f/m^3]
HELP1	:	hulpvariabele [f/m^3]
HELP2	:	,, [f/m^3]
I	:	index van de winputten [-]
I2	:	hulpindex [-]
IH1,IH2	:	hulpindices [-]
IND	:	0-1 variabele die aangeeft of een U0(J) in de huidige iteratiestap van TRANS gewaardeerd is [-]
IU2(I)	:	hulpvariabele [-]
J	:	index van de vraagpunten [-]
J1(J)	:	verzameling indices J van de variabelen U0(J) die nog niet gewaardeerd zijn [-]
J2	:	hulpindex [-]
JH,JH1,JH2	:	hulpindices [-]
JU2(J)	:	hulpvariabele [-]
K	:	index van de spline functie intervallen [-]
KQ(I,K)	:	zandproduktie van winput I waarbij het K-de knikpunt in de spline functie optreedt [m^3]
KS	:	$\sum_{K=1}^{KU} D4(I,K)$
KU	:	aantal spline functie intervallen [-]
M	:	aantal vraagpunten [-]
N	:	aantal winputten [-]
ND	:	totaal aantal dummievariabelen
NJ1	:	aantal indices J1(J) [-]
NJ2	:	aantal indices J1(J) [-]
NNV(J)	:	aantal winputten waaruit zand vervoerd wordt naar vraagpunt J [-]
NTP	:	totaal aantal variabelen van het volledige primale probleem (inclusief de spelingsvariabelen) [-]

NV(J,I)	:	nummer van de winputten waaruit zand vervoerd wordt naar vraagpunt J [-]
NVL	:	aantal variabelen U0(J) dat nog niet gewaardeerd is minus het aantal vergelijkingen dat nog niet gebruikt is om een variabele U0(J) of Z(I) te waarderen [-]
PSI(I,J)	:	hulpvariabele [f/m^3]
Q(I,K)	:	QA - KQ(I,K) [m^3]
QA	:	totale vraag naar zand [m^3]
QBU(I,S)	:	maximale hoeveelheid zand die vanuit winput I getransporteerd wordt door leidingtraject S [m^3]
QX	:	hulpvariabele [m^3]
RQ0(I)	:	minimaal te produceren hoeveelheid zand ingeval winput I in produktie genomen wordt [m^3]
S	:	index van de leidingtrajecten [-]
SI(I)	:	aantal trajecten van het leidingennet van winput I [-]
SU	:	aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
T	:	index van de iteratiestappen van het oplossingsalgorithme [-]
TU	:	maximum aantal iteratiestappen van het oplossingsalgorithme [-]
SUM	:	hulpvariabele [f/m^3]
U0(J)	:	duale variabele [f/m^3]
U4(I,K)	:	duale spelingsvariabele [f/m^3]
U5(I,K)	:	duale spelingsvariabele [f/m^3]
X(J)	:	primale variabele [m^3]
X1(J)	:	dummievariabele [-]
YS0(J)	:	primale spelingsvariabele [m^3]
YS1(I)	:	primale spelingsvariabele [m^3]
Z(I)	:	duale variabele [f/m^3]

Uitvoerbestand van de subroutinetest

22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41

Gebruikte parameters:

KU: 5 M : 5 N: 2 SU: 25 TU: 1

ND: 64 NTP: 37

Tussenuitkomsten:

NNV(1) : 1 NNV(2) : 1 NNV(3) : 1 NNV(4) : 1 NNV(5) : 1

NV(1,1): 2 NV(2,1): 1 NV(3,1): 1 NV(4,1): 1 NV(5,1): 1

Stap 1:

I : 1 QX : 0.0 YS1(I): 110000.0

I : 2 QX : 211892.9 YS1(I): 211892.9

GI(1) : 1 Z(1) : 0.000

GI(2) : 1 Z(2) : 0.000

Stap 2:

YS0(1) : 0.0

YS0(2) : 0.0

YS0(3) : 0.0

YS0(4) : 0.0

YS0(5) : 0.0

GI(1) : 1 Z(1) : 0.0

GI(2) : 1 Z(2) : 0.0

U0(1) : 2.285

U0(2) : 7.666

U0(3) : 2.029

U0(4) : 2.398

U0(5) : 4.516

NJ1 : 0

Stap 6:

Z(1) : 0.000 Z(2) : 0.000

U0(1) : 2.285

U0(2) : 7.666

U0(3) : 2.029

U0(4) : 2.398

U0(5) : 4.516

Berekende variabelen:

BE(1) : 2949491.3

GAM(1,J):	148435.8	151989.5	15000.0	15000.0	52620.0
	67920.0	166630.0	215080.0	0.0	135840.0
	0.0	49616.1	0.0	151496.1	0.0
	21316.1	39465.0	50940.0	48235.0	62260.0
	92085.0	118860.0	0.0	202436.1	65775.0
	0.0	83315.0	0.0	39465.0	0.0
	-41693.0	21316.0	-46078.0	15656.1	-37308.0
	26976.1	-46078.0	0.0	-46078.0	0.0
	-28538.0	0.0	-6613.0	0.0	-24153.0
	0.0	-63618.0	0.0	26310.0	0.0
	48235.0	62260.0	-27438.0	56600.0	-488383.3
	95010.5	-31909.7	50463.6	9267.4	18854.1
	59279.1	72436.1	142842.9	163566.8	

DUOPT : 3705520.5

Controle d.m.v. handberekening

Bepalen NNV(J) en NV(J,JH)

```
J=1, I=1: IH1=1, IH2=11                J=5, I=1: IH1=5, IH2=15
          X(1)+X(11)=0.0                         X(5)+X(15)=40000.0>0.0
     I=2: IH1=6, IH2=16                          NNV(5)=1, NV(5,1)=1
          X(6)+X(16)=1000000.0>0.0          I=2: IH1=10, IH2=20
          NNV(1)=1, NV(1,1)=2                    X(10)+X(20)=0.0
```

Stap 1:

```
I=1: Z(1)=0.0, GI(1)=1
     D1(1)>0, KS=0
     QX=X(20)=0.0, YS1(1)=X(36)=110000.0
     (1-D4(1,1))*YS1(1)+D4(1,1)*QX=110000.0>0
     Z(1)=0.0, GI(1)=1

I=2: Z(2)=0.0, GI(2)=1
     D1(2)>0, KS=2
     QX=X(27)=211892.9, YS1(2)=X(37)=211892.9
     (1-D4(2,1))*YS1(2)+D4(2,1)*QX=423965.8>0
     Z(1)=0.0, GI(1)=1
```

Stap 2:

```
YS0(1-5)=X(31-35)=0.0

NJ1=0

J=1: I=2, GI(2)=1, U0(1)=PSI(2,1)-Z(2)=2.285
     I=2, Z(2)=PSI(2,1)-U0(1)=0.0, GI(2)=1

J=2: I=1, GI(1)=1, U0(2)=PSI(1,2)-Z(1)=7.666
     I=1, Z(1)=PSI(1,2)-U0(2)=0.0, GI(1)=1
```

J=3-5: idem

Stap 3:

NJ1=0: naar stap 6

Stap 6:

$$BE(1)=\sum_{J=1}^{5} U0(J)*A(J)$$
=1000000.0*2.285+40000.0*(7.666+2.029+2.398+4.516)=2949360.0
≈2949491.3

I=1: KS=0, U4(1,1-5)=0.0, U5(1,1-5)=-H(1,1-5)=
U5(1,1-5)=0.491, 0.066, 0.045, 0.032, 0.022

I=2: KS=2, U4(2,1-5)=0.0, U5(2,1-5)=-H(2,1-5)=
U5(2,1-5)=0.533, 0.064, 0.044, 0.031, 0.022
U4(2,1-2)=-H(2,1-2)=0.533, 0.064
U5(2,1-2)=0.0
U4(2,2)=U4(2,2)+Z(2)=0.064+0.0=0.064

I=1: HELP1=0.0, D1(1)=1>0:
GAM(1,1)=148435.8+50000.0*(1-0)*0.0-0.0*1160000.0=148435.8
HELP1=0.0, SUM=0.0, D2(1)=1>0:
GAM(1,3)=15000.0-0.0*1160000.0=15000.0

I=1, S=1: HELP1=0.0, D3(1,1)=0:
HELP1=1*(2.285-5.720-0.0)=-3.435
J=2: HELP2=0*(7.666-5.418-0.0)=0.0
0.0>-3.435: HELP1=0.0
J=3-5: idem
JH=2*2+(1-1)*2+1=5
GAM(1,5)=52620.0-0.0*1000000=52620.0

I=1, S=13: HELP1=0.0, D3(1,13)=1>0:
JH=2*2+(13-1)*2+1=29
GAM(1,29)=39465.0-0.0*160000=39465.0

I=1, K=1: JH=2*2+2*25+(1-1)*2+1=55
GAM(1,55)=164678.4*0.0-995321.6*0.491=-488702.9≈488383.3

I=2: HELP1=0.0, D1(2)=1>0:
GAM(1,2)=151989.5+50000.0*(1-1)*0.0-0.0*1160000.0=151989.5

HELP1=0.0, SUM=1*-0.533+1*-0.064=-0.597
D2(2)=0: HELP1=2.285-5.245+0.597=-2.363
J=2: HELP2=7.666-9.445+0.597=-1.182
HELP2>HELP1: HELP1=-1.182
J=3: HELP2=2.029-7.695+0.597=-5.069
HELP2<HELP1: HELP1=-1.182
J=4: HELP2=2.398-7.695+0.597=-4.700
HELP2<HELP1: HELP1=-1.182

```
J=5: HELP2=4.516-7.345+0.597=-2.232
     HELP2<HELP1: HELP1=-1.182

HELP1=-1.182<0.0: HELP1=0.0
GAM(1,4)=15000.0-0.0*1160000.0=15000.0
```

$$\text{DUOPT}= \text{BE(T)}+ \sum_{I=1}^{2} \text{GAM}(1,I)*\text{D1}(I)+ \sum_{I=1}^{2} \text{GAM}(1,N+I)*\text{D2}(I)$$

$$+ \sum_{I=1}^{2} \sum_{S=1}^{25} \text{GAM}(1,2*2+I*S)*\text{D3}(I,S) + \sum_{I=1}^{2} \sum_{K=5}^{5} \text{GAM}(1,2*2+2*25+I*K)$$

```
= 2949360.0
  +  148435.8*1+151989.5*1+ 15000.0*1+ 15000.0*0+ 52620.0*0
  +   67920.0*1+166630.0*0+215080.0*0+     0.0*0+135840.0*0
  +       0.0*0+ 49616.1*0+     0.0*0+151496.1*0+     0.0*0
  +   21316.1*0+ 39465.0*0+ 50940.0*1+ 48235.0*0+ 62260.0*1
  +   92085.0*0+118860.0*0+     0.0*0+202436.1*0+ 65775.0*0
  +       0.0*0+ 83315.0*0+     0.0*0+ 39465.0*1+     0.0*0
  -   41693.0*0+ 21316.0*0+-46078.0*0+ 15656.1*0 -37308.0*0
  +   26976.1*0 -46078.0*0+     0.0*0 -46078.0*0+     0.0*0
  -   28538.0*0+     0.0*0  -6613.0*0+     0.0*0 -24153.0*0
  +       0.0*0 -63618.0*0+     0.0*0+ 26310.0*1+     0.0*0
  +   48235.0*1+ 62260.0*0 -27438.0*0+ 56600.0*0-488383.3*0
  +   95010.5*1 -31909.7*0+ 50463.6*1+  9267.4*0+ 18854.1*0
  +   59279.1*0+ 72436.1*0+142842.9*0+163566.8*0
= 3705389.4≈3705520.5
```

Het verschil van 0.0035 % is hierbij te wijten aan afrondingsfouten van de bij de handberekening gebruikte waarden.

BIJLAGE XXV

Listing van het programma TESTZEON

```
      PROGRAM TESTZEON
C
C     Declareren van de parameters
C
      INTEGER ND,TU
C
C     Definiëren van de parameters
C
      PARAMETER (ND=7,TU=5)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER X1(ND),TZ
      REAL BE(TU),GAM(TU,ND),Z
C
C     Declareren van de indices
C
      INTEGER J,T
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*47 PROBLEM
C
C     Communiceren met de subroutine ZEON
C
      COMMON/BLOCK1/ BE,GAM,X1,Z
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) PROBLEM
      WRITE(2,25) DATE,PROBLEM
C
      TZ=TU
C
C     Inlezen van de invoergegevens
C
      READ(1,*) BE,((GAM(T,J),J=1,ND),T=1,TZ)
C
C     Uitvoeren van de gebruikte parameters en ingelezen gegevens
C
      WRITE(2,35)
      WRITE(2,45) ND,TU
      WRITE(2,55)
      WRITE(2,65) (BE(T),T=1,TZ)
      WRITE(2,75) (GAM(T,J),J=1,ND),T=1,TZ)
C
C     Aanroepen van de subroutine ZEON
C
      CALL ZEON(TZ)
```

```
C
C     Uitvoeren van de berekende variabelen
C
      WRITE(2,85)
      WRITE(2,95) (X1(J),J=1,ND)
      WRITE(2,105) Z
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A47)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,'Gebruikte parameters:')
   45 FORMAT(/X,'NC:',X,I2,3X,'NV:',X,I2)
   55 FORMAT(//X,'Ingelezen gegevens:')
   65 FORMAT(/X,'BE(T)',3X,':',X,5(X,F5.1))
   75 FORMAT(/X,'GAM(T,J):',X,7(X,F5.1)/(10X,7(X,F5.1)))
   85 FORMAT(/X,''Berekende waarden:')
   95 FORMAT(/X,'X1(J)',3X,':',X,7(3X,I1))
  105 FORMAT(/X,'Z',7X,':',2X,F5.1)
      END
```

Listing van de subroutine ZEON

```
      SUBROUTINE ZEON(TZ)
C
C     Declareren van de parameters
C
      INTEGER ND,TU
C
C     Definiëren van de parameters
C
      PARAMETER (ND=7,TU=5)
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER X1(ND),TZ
      REAL BE(TU),GAM(TU,ND),Z
C
C     Declareren van de lokale variabelen
C
      INTEGER G(ND),GN,IND,J1,L1,M,MMAX,T1,XS(ND)
      REAL AH(0:1),AL(TU),AM(0:1),AS,FI(TU),IN(ND),MAX,MIN
C
C     Declareren van de indices
C
      INTEGER J,T
C
C     Communiceren met de subroutine ZEON
C
      COMMON/BLOCK1/ BE,GAM,X1,Z
C
C     Stap 0
C
```

```
      M=0
      GN=0
      AS=BE(1)
      DO 10 T=2,TZ
            IF(BE(T).GT.AS)AS=BE(T)
   10 CONTINUE
      DO 40 J=1,ND
            G(J)=1
            IND=0
            XS(J)=0
            X1(J)=0
            DO 20 T=1,TZ
                  FI(T)=0.0
                  IF(GAM(T,J).LT.0.0)IND=1
   20       CONTINUE
            IF(IND.EQ.0)THEN
              G(J)=0
              GN=GN+1
              GOTO 50
            ENDIF
            IND=0
            DO 30 T=1,TZ
                  IF(GAM(T,J).GT.0.0)GOTO 40
                  IF(GAM(T,J).LT.0.0)IND=1
   30       CONTINUE
            IF(IND.EQ.1)THEN
              X1(J)=1
              G(J)=0
              GN=GN+1
            ENDIF
   40 CONTINUE
      IF(TZ.EQ.1)GOTO 170
      MMAX=ND-GN
C
C     Stap 1
C
   50 CONTINUE
      M=M+1
      AL(1)=BE(1)
      DO 60 J=1,ND
            AL(1)=AL(1)+X1(J)*GAM(1,J)
   60 CONTINUE
      MAX=AL(1)
      T1=1
      DO 80 T=2,TZ
            AL(T)=BE(T)
            DO 70 J=1,ND
                  AL(T)=AL(T)+X1(J)*GAM(T,J)
   70       CONTINUE
            IF(AL(T).LE.MAX)GOTO 80
            MAX=AL(T)
            T1=T
   80 CONTINUE
      IF(MAX.LT.AS)THEN
        AS=MAX
        DO 90 J=1,ND
```

```
            XS(J)=X1(J)
   90   CONTINUE
      ENDIF
C
C     Stap 2
C
      MIN=0.0
      DO 100 J=1,ND
             PROD=G(J)*GAM(T1,J)
             IF(PROD.GE.MIN)GOTO 100
             MIN=GAM(T1,J)
             J1=J
             IN(M)=J1
  100 CONTINUE
      IF(MIN.EQ.0.0)GOTO 140
C
C     Stap 3
C
      G(J1)=0
      AH(0)=AL(1)
      DO 110 J=1,ND
             IF(GAM(1,J).LT.0.0)AH(0)=AH(0)+G(J)*GAM(1,J)
  110 CONTINUE
      AH(1)=AH(0)+GAM(1,J1)
      AM(0)=AH(0)
      AM(1)=AH(1)
      DO 130 T=2,TZ
             AH(0)=AL(T)
             DO 120 J=1,ND
                    IF(GAM(T,J).LT.0.0)AH(0)=AH(0)+G(J)*GAM(T,J)
  120        CONTINUE
             AH(1)=AH(0)+GAM(T,J1)
             IF(AH(0).GT.AM(0))AM(0)=AH(0)
             IF(AH(1).GT.AM(1))AM(1)=AH(1)
  130 CONTINUE
      L1=1
      IF(AM(0).LT.AM(1))L1=0
      X1(J1)=L1
      FI(J1)=AM(1-L1)
      IF(AM(L1).LT.AS)GOTO 50
C
C     Stap 4
C
  140 CONTINUE
      M=M-1
      IF(M.EQ.0)GOTO 150
      J1=IN(M)
      IF(FI(J1).GE.AS)THEN
        G(J1)=1
        X1(J1)=0
        GOTO 140
      ENDIF
      X1(J1)=1-X1(J1)
      FI(J1)=AS+1.0
      GOTO 50
```

```
C
C     Stap 5
C
  150 CONTINUE
      Z=AS
      DO 160 J=1,ND
             X1(J)=XS(J)
  160 CONTINUE
      RETURN
C
C     Bepalen van de optimale oplossing wanneer er maar één
C     vergelijking is
C
  170 CONTINUE
      Z=BE(1)
      DO 180 J=1,ND
             Z=Z+X1(J)*GAM(1,J)
  180 CONTINUE
      RETURN
      END
```

Betekenis van de in het programma TESTZEON en de subroutine ZEON gebruikte variabelen

```
AH(0:1) : hulpvariabele [-]
AL(T)   : de waarde van beperking T [-]
AS      : de minimumwaarde van Z die tot dan toe bereikt is [-]
BE(T)   : constante van de beperking T [-]
DATE    : datum van de programma-run [-]
G(J)    : variabele die aangeeft welke dummievariabele X1(J)
          gewaardeerd is [-]
GAM(T,J): coëfficiënten van de dummievariabelen in beperking T
          [-]
J       : index van de dummievariabelen [-]
ND      : totaal aantal dummievariabelen.
MAX     : hulpvariabele [-]
MIN     :       ,,      [-]
PROBLEM : identificatie van het testprobleem [-]
T       : index van de beperkingen [-]
TU      : aantal beperkingen [-]
TZ      : aantal beperkingen [-]
X1(J)   : dummievariabele
XS(J)   : dummievariabele behorende bij AS [-]
Z       : waarde van de oplossing van ZEON [-]
```

Invoerbestand van het testprobleem

```
22-10-1986
Testprobleem: aangepast voorbeeld ICW-nota 1335
10.0, 13.0,  9.0, 15.0, 11.0,
-8.0, 15.0,  6.0,  2.0,-11.0,  1.0, -1.0,
 2.0,-10.0,  3.0, -1.0, -6.0,  1.0,  0.0,
-5.0, -5.0,  8.0,  2.0,  0.0,  1.0, -1.0,
-3.0, -6.0, 11.0, -3.0, -4.0,  1.0,  0.0,
 8.0,  1.0,-10.0,  1.0,  1.0,  1.0, -1.0
```

Uitvoerbestand van het testprobleem

22-10-1986
Testprobleem: aangepast voorbeeld ICW-nota 1335

Gebruikte parameters:

ND: 7 TU: 5

Ingelezen gegevens:

BE(T) :	10.0	13.0	9.0	15.0	11.0		
GAM(T,J):	-8.0	15.0	6.0	2.0	-11.0	1.0	-1.0
	2.0	-10.0	3.0	-1.0	-6.0	1.0	0.0
	-5.0	-5.0	8.0	2.0	0.0	1.0	-1.0
	-3.0	-6.0	11.0	-3.0	-4.0	1.0	0.0
	8.0	1.0	-10.0	1.0	1.0	1.0	-1.0

Berekende waarden:

X1(J) :	0	0	0	0	1	0	1

Z : 11.0

BIJLAGE XXVI

Listing van het hoofdprogramma MAIN

```
      PROGRAM MAIN
C
C     Declareren van de parameters
C
      INTEGER KU,M,N,NCP,ND,NTP,NVP,SU,TU
C
C     Toevoegen van het parameterbestand
C
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(M),AM(NCP,NVP),B(N),COEF(N,SU,M),I2,IU2(N),J2,
     #        JU2(M),NC,NV,QA,QBU(N,SU),XB(NCP),X1(ND)
      REAL AB,AU(N),BE(TU),BI(NCP,NCP),BL(NCP),C(NVP),DUOPT,E(N,SU),
     #     F(N,M),G(N,M),GAM(TU,ND),H(N,KU),KQ(N,KU),PRIOPT,PSI(N,M),
     #     Q(N,KU),RES,RQ0(N),TCG,U0(M),X(NTP),XR(NVP),Z(N),ZR,ZZ
C
C     Declareren van de indices
C
      INTEGER I,J,T
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*62 AREA
C
C     Communiceren met de subroutines ANSOL, PRISOL, RESIM, TRANS
C     en ZEON
C
      COMMON/BLOCK1/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK2/ Q,QA,QBU
      COMMON/BLOCK3/ D1,D2,D3,D4,TCG
      COMMON/BLOCK4/ PRIOPT,PSI,RES,X
      COMMON/BLOCK5/ AM,BI,BL,C,NC,NV,XB,XR,ZR
      COMMON/BLOCK6/ AB,I2,IU2,J2,JU2,U0,Z
      COMMOM/BLOCK7/ BE,GAM,DUOPT,ZZ
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) AREA
      WRITE(2,25) DATE,AREA
C
C     Uitvoeren van de gebruikte parameters
C
      WRITE(2,35)
      WRITE(2,45) KU,M,N,SU,TU,NCP,ND,NTP,NVP
C
C     Aanroepen van de subroutines RECALCO en ANSOL
C
```

```
      CALL RECALCO
      CALL ANSOL
C
C     Uitvoeren van het in ANSOL berekende minimum TCG
C
      WRITE(2,55)
      WRITE(2,65) TCG
      T=0
      ZZ=0.0
   10 CONTINUE
C
C     De teller die het aantal iteraties bijhoudt met één verhogen
C
      T=T+1
C
C     Controleren of de teller het vooraf vastgestelde maximum aantal
C     iteraties overschrijdt
C
      IF(T.GT.TU) THEN
        WRITE(2,75)
        STOP
      ENDIF
C
C     Het nummer van de iteratie uitvoeren
C
      WRITE(2,85) T
C
C     Aanroepen van de subroutine PRISOL
C
      CALL PRISOL(*20,*40)
C
C     Uitvoeren van de in de subroutine PRISOL berekende minima
C
   20 CONTINUE
      WRITE(2,95) ZR
      WRITE(2,105) RES
      WRITE(2,115) PRIOPT
C
C     Testen of de optimale oplossing bereikt is en zo ja,
C     de berekende waarden uitvoeren en het programma beeindigen
C
      IF(ABS(PRIOPT-ZZ)/PRIOPT.LT.0.001)THEN
        WRITE(2,125) T
        DO 30 I=1,N
              WRITE(2,135) I, (X(J),J=(I-1)*M+1,I*M)
              WRITE(2,145) I, (X(J),J=N*M+(I-1)*M+1,N*M+I*M)
              WRITE(2,155) I, (X(J),J=2*N*M+(I-1)*KU+1,2*N*M+I*KU)
   30   CONTINUE
        WRITE(2,165) (X(J),J=2*N*M*+N*KU+1,2*N*M+N*KU+M)
        WRITE(2,175) (X(J),J=NTP-N+1,NTP)
        WRITE(2,185) PRIOPT
        STOP
      ENDIF
C
C     Aanroepen van de subroutine TRANS
C
```

```
      CALL TRANS(T)
C
C     Uitvoeren van het in de subroutine TRANS berekende maximum
C
      WRITE(2,195) DUOPT
C
C     Aanroepen van de subroutine ZEON
C
      CALL ZEON(T)
C
C     Uitvoeren van het in de subroutine ZEON berekende minimum
C
      WRITE(2,205) ZZ
C
C     Itereren
C
      GOTO 10
   40 CONTINUE
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
   15 FORMAT(A62)
   25 FORMAT(X,A/X,A)
   35 FORMAT(//X,Gebruikte parameters:')
   45 FORMAT(/X,'KU',X,':',X,I2,5X,'M',X,':',X,I2,5X,'N',2X,':',X,I2,
     #       5X,'SU',X,':',X,I2,5X,'TU:',X,I2//X,'NCP:',X,I2,5X,'ND:',
     #       X,I2,5X,'NTP:',X,I2,5X,'NVP:',X,I2)
   55 FORMAT(//X,'Berekende variabelen:')
   65 FORMAT(/X,'TCG',3X,':',X,F10.1)
   75 FORMAT(//X,'Het hoofdprogramma MAIN is gestopt omdat het ',
     #          'maximum aantal'/X,'iteraties bereikt is')
   85 FORMAT(//X,'Iteratie',X,I2':')
   95 FORMAT(/X,'ZR',4X,':',X,F10.1)
  105 FORMAT(/X,'RES',3X,':',X,F10.1)
  115 FORMAT(/X,'PRIOPT:',X,F10.1)
  125 FORMAT(//X,'Het in iteratie',X,I2,X,'bereikte optimum:')
  135 FORMAT(/X,'QW(',I1,',J):',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  145 FORMAT(/X,'QB(',I1,',J):',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  155 FORMAT(/X,'QX(',I1,',K):',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  165 FORMAT(/X,'YS0(J)',2X,':',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  175 FORMAT(/X,'YS1(I)',2X,':',2X,F9.1,4(3X,F9.1)/(8X,5(3X,F9.1)))
  185 FORMAT(/X,'TCG',3X,':',X,F10.1)
  195 FORMAT(/X,'DUOPT',X,':',X,F10.1)
  205 FORMAT(/X,'ZZ',4X,':',X,F10.1)
      END
```

Listing van de subroutine RECALCO

De listing van de subroutine RECALCO is identiek aan die van bijlage XIX zonder de PARAMETER-statement en met uitzondering van de volgende INCLUDE- en COMMON-statements:

```
      INCLUDE 'PROGRAM.PAR'
C
C     Communiceren met de subroutines ANSOL, PRISOL en TRANS
C
      COMMON/BLOCK1/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK2/ Q,QA,QBU
```

Listing van de subroutine ANSOL

De listing van de subroutine ANSOL is identiek aan die van bijlage XX zonder de PARAMETER-statement, de WRITE- en FORMAT-statements t.b.v. het uitvoeren van de tussenuitkomsten en met uitzondering van de volgende INCLUDE- en COMMON-statements:

```
      INCLUDE 'PROGRAM.PAR'
C
C     Communiceren met het hoofdprogramma MAIN en de subroutines
C     RECALCO, PRISOL en TRANS
C
      COMMON/BLOCK1/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK2/ Q,QA,QBU
      COMMON/BLOCK3/ D1,D2,D3,D4,TCG
```

Listing van de subroutine PRISOL

De listing van de subroutine PRISOL is identiek aan die van bijlage XXII zonder de PARAMETER-statement, de WRITE- en FORMAT-statements t.b.v. het uitvoeren van de tussenuitkomsten en met uitzondering van de volgende INCLUDE- en COMMON-statements:

```
      INCLUDE 'PROGRAM.PAR'
C
C     Communiceren met het hoofdprogramma MAIN en de subroutines
C     RECALCO, ANSOL, RESIM en TRANS
C
      COMMON/BLOCK1/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK3/ D1,D2,D3,D4
      COMMON/BLOCK4/ PRIOPT,PSI,RES,X
      COMMON/BLOCK5/ AM,BL,BI,C,NC,NV,XB,XR,ZR
```

Listing van de subroutine RESIM

De listing van de subroutine RESIM is identiek aan die van bijlage XXI zonder de PARAMETER-statement en met uitzondering van de volgende INCLUDE-, declaratie-, COMMON- en FORMAT-statements:

```
      INCLUDE 'PROGRAM.PAR'
C
C     Declareren van de niet-lokale variabelen
C
      INTEGER A(NCP,NVP),XB(NCP)
      REAL B(NCP),BI(NCP,NCP),C(NVP),X(NVP),Z
```

```
C
C     Declareren van de lokale variabelen
C
      INTEGER IND,K,R
      REAL AA(NCP),BA,FRAC,HELP,HELP1,HELP2(NCP),PROD(NCP),SUM
C
C     Communiceren met het hoofdprogramma MAIN en de subroutine PRISOL
C
      COMMON/BLOCK5/ A,BI,B,C,NC,NV,XB,X,Z
    5 FORMAT(////X,'Het hoofdprogramma MAIN is gestopt omdat de ',
     #       'startbasis van de'/X,'subroutine RESIM niet toelaat',
     #       'baar is')
   15 FORMAT(////X,'Het hoofdprogramma MAIN is gestopt omdat een ',
     #       'van de variabelen'/X,'in de subroutine RESIM',
     #       ' onbegrensd is')
```

Listing van de subroutine TRANS

De listing van de subroutine TRANS is identiek aan die van bijlage XXIV zonder de PARAMETER-statements en de WRITE- en FORMAT-statements t.b.v. het uitvoeren van tussenuitkomsten en met uitzondering van de volgende INCLUDE- en COMMON-statements:

```
      INCLUDE 'PROGRAM.PAR'
C
C     Communiceren met het hoofdprogramma MAIN en de subroutines
C     RECALCO, ANSOL, PRISOL, CORRECT en ZEON
C
      COMMON/BLOCK1/ A,AU,B,COEF,E,F,G,H,KQ,RQ0
      COMMON/BLOCK2/ Q,QA,QBU
      COMMON/BLOCK3/ D1,D2,D3,D4,TCG
      COMMON/BLOCK4/ PRIOPT,PSI,RES,X
      COMMON/BLOCK6/ AB,I2,IU2,J2,JU2,U0,Z
      COMMON/BLOCK7/ BE,GAM,DUOPT,ZZ
```

Listing van de subroutine CORRECT

De listing van de subroutine CORRECT is identiek aan die van bijlage XXIII zonder de PARAMETER-statement en met uitzondering van de volgende INCLUDE- en COMMON-statement:

```
      INCLUDE 'PROGRAM.PAR'
C
C     Communiceren met de subroutine TRANS
C
      COMMON/BLOCK7/ BE,GAM,DUOPT,ZZ
```

Listing van de subroutine ZEON

De listing van de subroutine ZEON is identiek aan die van bijlage XXV zonder de PARAMETER-statement en met uitzondering van de volgende INCLUDE- en COMMON-statements:

```
      INCLUDE 'PROGRAM.PAR'
C
C     Communiceren met het hoofdprogramma MAIN en de subroutines
C     PRISOL en TRANS
C
      COMMON/BLOCK3/ D1,D2,D3,D4,TCG
      COMMON/BLOCK7/ BE,GAM,DUOPT,ZZ
```

Betekenis van de in het programma MAIN gebruikte variabelen

AREA : identificatie van het proefgebied [-]
DATE : datum van de programma-run [-]
DUOPT : maximum van het duale probleem, zoals berekend in de subroutine TRANS [f]
I : index van de winputten [-]
J : index van de dummievariabelen [-]
KU : aantal spline functie intervallen [-]
M : aantal vraagpunten [-]
N : aantal winputten [-]
NCP : aantal beperkingen van het volledige primale probleem [-]
ND : totaal aantal dummievariabelen [-]
NTP : totaal aantal variabelen van het volledige primale probleem (inclusief de spelingsvariabelen) [-]
NVP : aantal variabelen van het volledige primale probleem [-]
PRIOPT: minimum van het primale probleem, zoals berekend in de subroutine PRISOL [f]
RES : bijdrage van de dummievariabelen aan PRIOPT [f]
SU : aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
T : index van de iteratiestappen van het oplossingsalgorithme [-]
TCG : minimum van het primale probeem, zoals berekend in de subroutine ANSOL [f]
TU : maximum aantal iteratiestappen van het oplossingsalgorithme [-]
X(J) : continue variabelen [m^3]
ZR : bijdrage van de continue variabelen aan PRIOPT [f]
ZZ : minimum van het primale probleem, zoals berekend in de subroutine ZEON [f]

De betekenis van de overige variabelen wordt verklaard in de subroutines, waarin deze variabelen daadwerkelijk gebruikt worden.

Listing van het parameterbestand PROGRAM.PAR van de oorspronkelijke versie

```
      PARAMETER(KU=5,M=5,N=2,SU=25,TU=25)
      PARAMETER(NCP=N+M,ND=2*N+N*SU+N*KU)
      PARAMETER(NTP=2*N*M+N+M+N*KU,NVP=N*M+N+M)
```

Uitvoerbestand van de programmarun van de oorspronkelijk versie

22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41

Gebruikte parameters:

KU : 5 M : 5 N : 2 SU : 25 TU: 25

NCP: 7 ND: 64 NTP: 37 NVP: 17

Berekende variabelen:

TCG : 3647040.3

Iteratie 1:

ZR : 2949491.0

RES : 756029.4

PRIOPT: 3705520.5

DUOPT : 3705520.5

ZZ : 2291942.0

Iteratie 2:

ZR : 7037209.5

RES : 797432.6

PRIOPT: 7834642.0

DUOPT : 7834642.0

ZZ : 2470241.5

Iteratie 3:

Het hoofdprogramma MAIN is gestopt omdat de startbasis van de subroutine RESIM niet toelaatbaar is

BIJLAGE XXVII

Listing van het programma TESTMODIFY

```
      PROGRAM TESTMODIFY
C
C     Declareren van de parameters
C
      INTEGER CO,KU,M,N,SU
C
C     Definiëren van de parameters
C
      PARAMETER (KU=5,M=5,N=2,SU=86,CO=2*N+N*(SU+KU))
C
C     Declareren van de variabelen
C
      INTEGER COEF(N,SU,M),COMB(CO,0:25),COMBMAX,DT(N,SU),DTO(SU),
     #        END(N,M),G(SU,0:M),ND,SI(N),SIMAX,SIVAR1,SIVAR2
C
C     Declareren van de indices
C
      INTEGER I,J,JD,JH,JH1,JH2,K,S,SH,SH1
C
C     Declareren van de variabelen t.b.v. de identificatie
C
      CHARACTER*10 DATE
      CHARACTER*62 AREA
C
C     Inlezen en uitvoeren van de variabelen t.b.v. de identificatie
C
      READ(1,5) DATE
      READ(1,15) AREA
      WRITE(2,25) DATE,AREA
C
C     Inlezen van de invoergegevens
C
      READ(1,*) (((COEF(I,S,J),S=1,SU),J=1,M),I=1,N)
      READ(1,*) ((END(I,J),I=1,N),J=1,M)
      READ(1,*) (DTO(S),S=1,SU)
C
C     Uitvoeren van de gebruikte parameters
C
      WRITE(2,35)
      WRITE(2,45) KU,M,N,SU,CO
      WRITE(2,55)
C
      JD=0
      SIMAX=0
      COMBMAX=0
      DO 210 I=1,N
             SI(I)=0
C
C     Bepalen van de afzonderlijke leidingnetten per winput
C
             DO 30 S=1,SU
```

```
                DO 20 J=1,M
                      IF(COEF(I,S,J).EQ.1)THEN
                        SI(I)=SI(I)+1
                        SH=SI(I)
                        DT(I,SH)=DTO(S)
                        DO 10 JH=1,M
                              COEF(I,SH,JH)=COEF(I,S,JH)
                              IF(END(I,JH).EQ.S)END(I,JH)=SH
   10                   CONTINUE
                        GOTO 30
                      ENDIF
   20               CONTINUE
   30         CONTINUE
C
C     Berekenen van G(S,0) en G(S,JH)
C
              DO 50 S=1,SI(I)
                    G(S,0)=0
                    DO 40 J=1,M
                      IF(COEF(I,S,J).EQ.1)THEN
                        G(S,0)=G(S,0)+1
                        JH=G(S,0)
                        G(S,JH)=J
                      ENDIF
   40               CONTINUE
   50         CONTINUE
C
C     Samenvoegen van trajecten
C
              SIVAR1=SI(I)
              SIVAR2=SI(I)
              DO 130 S=1,SI(I)-1
                     DO 120 SH=SIVAR1,S+1,-1
                        DO 60 JH=0,G(S,0)
                              IF(G(S,JH).NE.G(SH,JH))GOTO 120
   60                   CONTINUE
                        DT(I,S)=DT(I,S)+DT(I,SH)
                        DO 70 JH=1,G(SH,0)
                              JH1=G(SH,JH)
                              IF(END(I,JH1).EQ.SH)END(I,JH1)=S
   70                   CONTINUE
                        DO 110 SH1=SH,SIVAR2-1
                               DT(I,SH1)=DT(I,SH1+1)
                               DO 80 J=1,M
                                     COEF(I,SH1,J)=COEF(I,SH1+1,J)
   80                          CONTINUE
                               DO 90 JH=1,G(SH1+1,0)
                                     JH1=G(SH+1,JH)
                                     IF(END(I,JH1).EQ.SH1+1)
     #                                 END(I,JH1)=SH1
   90                          CONTINUE
                               DO 100 JH=0,G(SH1+1,0)
                                      G(SH1,JH)=G(SH1+1,JH)
  100                          CONTINUE
  110                   CONTINUE
                        SIVAR2=SIVAR2-1
```

```
  120                 CONTINUE
                      SIVAR1=SIVAR2
  130         CONTINUE
C
C     Bepalen van de combinatiematrix
C
              SI(I)=SIVAR2
              COMB(JD+1,0)=0
              DO 140 J=JD+2,JD+2+SI(I)+KU
                      COMB(J,0)=1
                      COMB(J,1)=JD+1
  140         CONTINUE
              DO 170 S=1,SI(I)
                      J=JD+2+S
                      DO 160 SH=1,SI(I)
                              IF(SH.EQ.S)GOTO 160
                              DO 150 JH1=1,G(S,0)
                                      JH2=G(S,JH1)
                                      IF(COEF(I,SH,JH2).EQ.0)GOTO 160
  150                         CONTINUE
                              COMB(J,0)=COMB(J,0)+1
                              JH=COMB(J,0)
                              COMB(J,JH)=JD+2+SH
  160                 CONTINUE
  170         CONTINUE
              DO 190 K=1,KU
                      J=JD+2+SI(I)+K
                      DO 180 KH=1,KU
                              IF(KH.EQ.K)GOTO 190
                              COMB(J,0)=COMB(J,0)+1
                              JH=COMB(J,0)
                              COMB(J,JH)=JD+2+SI(I)+KH
  180                 CONTINUE
  190         CONTINUE
              IF(SI(I).GT.SIMAX)SIMAX=SI(I)
C
C     Uitvoeren van de berekende variabelen
C
              WRITE(2,65) I,SI(I)
              WRITE(2,75) I,((COEF(I,S,J),S=1,SI(I)),J=1,M)
              WRITE(2,85) I,(DT(I,S),S=1,SI(I))
              WRITE(2,95) I,(END(I,J),J=1,M)
              WRITE(2,105) (COMB(J,0),J=JD+1,JD+2+SI(I)+KU)
              DO 200 J=JD+2,JD+2+SI(I)+KU
                      WRITE(2,115) (COMB(J,JH),JH=1,COMB(J,0))
                      IF(COMB(J,0).GT.COMBMAX)COMBMAX=COMB(J,0)
  200         CONTINUE
              JD=JD+2+SI(I)+KU
  210 CONTINUE
      ND=JD
      WRITE(2,125) ND,SIMAX,COMBMAX
      STOP
C
C     Format statements
C
    5 FORMAT(A10)
```

```
 15 FORMAT(A62)
 25 FORMAT(X,A/X,A)
 35 FORMAT(//X,'Gebruikte parameters:)
 45 FORMAT(/X,'KU:',X,I1,5X,'M:',X,I1,5X,'N:',X,I1,5X,'SU:',X,I2,
   #       5X,'CO:',X,I3)
 55 FORMAT(//X,'Berekende variabelen:)
 65 FORMAT(/X,'SI(',I1,')',6X,':',X,I1)
 75 FORMAT(/X,'COEF(',I1,',S,J):',6(X,I1)/(13X,6(X,I1)))
 85 FORMAT(/X,'DT(',I1,'S)',4X,':',X,I3,5(3X,I3))
 95 FORMAT(/X'END(',I1,',J)',3X,':',5(3,I1))
105 FORMAT(/X,'COMB(J,0)',2X,':',13(X,I1))
115 FORMAT(/X,'COMB(J,JH)',X,':',6(2X,I2)/(13X,6(2X,I2)))
125 FORMAT(/X,'ND:',X,I4,3X,'SIMAX:'X,I2,3X,'COMBMAX:',X,I2)
    END
```

Betekenis van de in het programma TESTMODIFY gebruikte variabelen

AREA	: identificatie van het proefgebied [-]
CO	: aantal dummievariabelen volgens de oorspronkelijke nummering [-]
COEF(I,S,J)	: 0-1 variabele die aangeeft of traject S deel uitmaakt van het leidingtracé dat winput I met vraagpunt J verbindt [-]
COMB(J,0)	: aantal dummievariabelen dat afhankelijk is van dummievariabele X1(J) [-]
COMB(J,JH)	: combinatiematrix die aangeeft welke dummievariabelen afhankelijk zijn van dummievariabele X1(J) [-]
COMBMAX	: maximum aantal afhankelijke dummievariabelen [-].
DATE	: datum van de programma-run [-]
DT(I,S)	: lengte van traject S van het leidingennet van winput I [hm]
DTO(S)	: lengte van traject S volgens de oorspronkelijke nummering [hm]
END(I,J)	: nummer van het traject van het leidingennet van winput I dat direct grenst aan vraagpunt J [-]
G(S,0)	: aantal vraagpunten waarbij traject S deel uitmaakt van het leidingtracé naar deze vraagpunten [-]
G(S,JH)	: vraagpunten waarbij traject S deel uitmaakt van het leidingtracé naar deze vraagpunten [-]
I	: index van de winputten [-]
J	: index van de vraagpunten [-]
JD	: index van de dummievariabelen [-]
JH,JH1,JH2	: hulpindices [-]
K	: index van de spline functie intervallen [-]
KU	: aantal spline functie intervallen [-]
M	: aantal vraagpunten [-]
N	: aantal winputten [-]
ND	: totaal aantal dummievariabelen [-]
S	: index van de leidingtrajecten [-]
SH,SH1	: hulpindices [-]
SI(I)	: aantal trajecten van het leidingennet van winput I [-]
SIMAX	: maximum aantal trajecten dat in een leidingennet voorkomt [-]

SU : aantal leidingtrajecten volgens de oorspronkelijke nummering [-]
SIVAR1 : aantal leidingtrajecten (variabel), gebruikt bij het samenvoegen van trajecten, die naar dezelfde vraagpunt(en) leiden [-]
SIVAR2 : idem [-]

Invoerbestand van de programmatest

```
22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41
17*0,1,7*0,1,18*0,3*1,5*0,2*1,32*0,
54*0,1,11*0,4*1,0,5*1,10*0,
54*0,1,21*0,1,9*0,
54*0,1,21*0,2*1,8*0,
54*0,1,21*0,3*1,7*0,
17*0,1,26*0,2*1,40*0,
39*0,2*1,2*0,1,3*0,1,28*0,3*1,17*0,
25*0,2*1,19*0,1,30*0,1,8*0,
25*0,2*1,19*0,1,39*0,
25*0,2*1,19*0,1,31*0,2*1,7*0,
18,18,
67,67,
77,78,
78,27,
79,79,
36,27,26, 9, 7,16, 5,10, 7, 5,11,10,13,10,11, 9, 7,12, 7,15,
 7, 9,10,33,19,38,24,50,12,15,62,38,15,20, 9,29,23, 8,32,16,
34,23,45,11, 9,11,21,43, 9, 4,44,18,15,19, 9,12,11,11,10,11,
 8, 2,10, 7, 2, 4,11,10,12,10, 5,10,14,19,15, 6, 6,11,10,13,
11,10, 7, 9,12,11
```

Uitvoerbestand van de programmatest

```
22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41
```

Gebruikte parameters:

KU: 5 M: 5 N: 2 SU: 86 CO: 192

Berekende variabelen:

SI(1) : 6

COEF(1,S,J): 1 0 0 0 0 0
0 1 1 0 0 0
0 1 0 1 0 0
0 1 0 1 1 0
0 1 0 1 1 1

```
DT(1,S)     : 125   9 107   6  11  10

END(1,J)    :   1   3   4   5   6

COMB(J,0)   : 0 1 1 1 2 2 3 4 1 2 3 4 5

COMB(J,JH)  :   1
                1
                1
                1   4
                1   4
                1   4   6
                1   4   6   7
                1
                1   9
                1   9  10
                1   9  10  11
                1   9  10  11  12

SI(2)       : 5

COEF(2,S,J): 1 0 0 0 0
             0 0 1 0 0
             0 1 0 1 0
             0 1 0 0 0
             0 1 0 0 1

DT(1,S)     :  32  83 137  11  10

END(1,J)    :   1   3   4   2   5

COMB(J,0)   : 0 1 1 1 1 2 2 1 2 3 4 5

COMB(J,JH)  :  14
               14
               14
               14
               14  17
               14  17
               14
               14  21
               14  21  22
               14  21  22  23
               14  21  22  23  24

ND: 25   SIMAX: 6   COMBMAX: 5
```

Controle d.m.v. handberekening

<u>Winputten, vraagpunten en trajecten volgens oude/nieuwe nummering</u>

```
Winput 4, vraagpunt  8:  18  26  45  46  47  53  54
       2             1    1   1   1   1   1   1   1
```

vraagpunt 30:	55	67	68	69	70	72	73	74	75	76
2	2	3	3	3	3	3	3	3	3	3

vraagpunt 39:	55	77		
3	2	4		

vraagpunt 40:	55	77	78	
4	2	5	5	

vraagpunt 41:	55	77	78	79
5	2	6	6	6

<u>Lengte trajecten</u>

Lengte traject 1: 12+38+9+11+21+15+19=125 hm

<u>Combinaties</u>

Nummering dummievariabelen:

D1(1) :	1	D4(1,1):	9
D2(1) :	2	D4(1,2):	10
D3(1,1):	3	D4(1,3):	11
D3(1,2):	4	D4(1,4):	12
D3(1,3):	5	D4(1,5):	13
D3(1,4):	6		
D3(1,5):	7		
D3(1,6):	8		

Combinaties:

COMB(1,0) :	0	COMB(1,JH):	-				
COMB(1,1) :	1	COMB(1,JH):	1				
COMB(1,2) :	1	COMB(1,JH):	1				
COMB(1,3) :	1	COMB(1,JH):	1				
COMB(1,4) :	2	COMB(1,JH):	1	4			
COMB(1,5) :	2	COMB(1,JH):	1	4			
COMB(1,6) :	3	COMB(1,JH):	1	4	6		
COMB(1,7) :	4	COMB(1,JH):	1	4	6	7	
COMB(1,8) :	1	COMB(1,JH):	1				
COMB(1,9) :	2	COMB(1,JH):	1	9			
COMB(1,10):	3	COMB(1,JH):	1	9	10		
COMB(1,11):	4	COMB(1,JH):	1	9	10	11	
COMB(1,12):	5	COMB(1,JH):	1	9	10	11	12

BIJLAGE XXVIII

In- en uitvoerbestanden van twee programmaruns met de laatste versie van het programma

<u>Testprobleem 1:</u>

Listing van het parameterbestand PROGRAM.PAR

```
PARAMETER(CO=6,KU=5,M=5,N=2,SU=6,TU=25)
PARAMETER(NCP=N+M,ND=2*N+N*SU+N*KU)
PARAMETER(NTP=2*N*M+N+M+N*KU,NVP=N*M+N+M)
```

Invoerbestand voor het hulpprogramma MODIFY

```
22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41
17*0,1,7*0,1,18*0,3*1,5*0,2*1,32*0,
54*0,1,11*0,4*1,0,5*1,10*0,
54*0,1,21*0,1,9*0,
54*0,1,21*0,2*1,8*0,
54*0,1,21*0,3*1,7*0,
17*0,1,26*0,2*1,40*0,
39*0,2*1,2*0,1,3*0,1,28*0,3*1,17*0,
25*0,2*1,19*0,1,30*0,1,8*0,
25*0,2*1,19*0,1,39*0,
25*0,2*1,19*0,1,31*0,2*1,7*0,
18,18,
67,67,
77,78,
78,27,
79,79,
36,27,26, 9, 7,16, 5,10, 7, 5,11,10,13,10,11, 9, 7,12, 7,15,
 7, 9,10,33,19,38,24,50,12,15,62,38,15,20, 9,29,23, 8,32,16,
34,23,45,11, 9,11,21,43, 9, 4,44,18,15,19, 9,12,11,11,10,11,
 8, 2,10, 7, 2, 4,11,10,12,10, 5,10,14,19,15, 6, 6,11,10,13,
11,10, 7, 9,12,11
```

Uitvoerbestand van het hulpprogramma MODIFY

```
22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41
  6
100000
011000
010100
010110
010111
 125  9 107   6  11  10
  1  3  4  5  6
  0  1  1  1  2  2  3  4  1  2  3  4  5
   1
   1
```

```
    1
    1   4
    1   4
    1   4   6
    1   4   6   7
    1
    1   9
    1   9  10
    1   9  10  11
    1   9  10  11  12
   5
 10000
 00100
 01010
 01000
 01001
   32  83 137  11  10
   1  3  4  2  5
   0  1  1  1  1  2  2  1  2  3  4  5
   14
   14
   14
   14
   14  17
   14  17
   14
   14  21
   14  21  22
   14  21  22  23
   14  21  22  23  24
```

ND: 25 SIMAX: 6 COMBMAX: 5

Invoerbestand voor het hoofdprogramma

```
22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41
43850,100000,17000,32000,28.5,0.3355,5.0,50000.0,4.0,2.0,
10,12, 1, 2, 3,
56600,100000,17000,32000,30.0,0.4150,5.0,50000.0,4.0,2.5,
 5,17,12,12,11,
1000000,4*40000
```

Uitvoerbestand van het hoofdprogramma

```
22-10-1986
Testgebied: winputten 4 en 18, vraagpunten 8, 30, 39, 40 en 41


Gebruikte parameters:

CO :  5     KU:  5     M  :  5     N  :  2     SU:  6     TU: 25

NCP:  7     ND: 25     NTP: 37     NVP: 17
```

Berekende variabelen:

TCG : 3647040.3

Iteratie 1:

ZR : 2949491.0

RES : 756029.4

PRIOPT: 3705520.5

DUOPT : 3705520.5

ZZ : 2291942.0

Iteratie 2:

ZR : 7037209.5

RES : 797432.6

PRIOPT: 7834642.0

DUOPT : 7834642.0

ZZ : 2470241.5

Iteratie 3:

ZR : 6976936.5

RES : 975732.1

PRIOPT: 7952668.5

DUOPT : 7952668.5

ZZ : 2514182.5

Iteratie 4:

ZR : 5888428.0

RES : 919889.5

PRIOPT: 6808317.5

DUOPT : 6808317.5

ZZ : 2625051.5

Iteratie 5:

ZR : 5191346.5

RES : 1130542.1

PRIOPT: 6321888.5

DUOPT : 6321888.5

ZZ : 3052663.8

Iteratie 6:

ZR : 3092749.8

RES : 672360.1

PRIOPT: 3765109.8

DUOPT : 3765109.8

ZZ : 3336698.3

Iteratie 7:

ZR : 4169639.8

RES : 348109.5

PRIOPT: 4517749.5

DUOPT : 4517749.5

ZZ : 3485134.0

Iteratie 8:

ZR : 3689105.5

RES : 496545.3

PRIOPT: 4185650.8

DUOPT : 4185651.0

ZZ : 3580512.0

Iteratie 9:

ZR : 3155700.3

RES : 631020.8

PRIOPT: 3786721.0

DUOPT : 3786721.3

ZZ : 3584144.0

Iteratie 10:

ZR : 3546927.5

RES : 610555.3

PRIOPT: 4157482.8

DUOPT : 4157483.0

ZZ : 3606822.0

Iteratie 11:

ZR : 3084230.3

RES : 657330.8

PRIOPT: 3741561.0

DUOPT : 3741561.3

ZZ : 3621514.5

Iteratie 12:

ZR : 3001741.0

RES : 1150679.4

PRIOPT: 4152420.5

DUOPT : 4152420.5

ZZ : 3627619.0

Iteratie 13:

ZR : 2942234.3

RES : 749415.8

PRIOPT: 3691650.0

DUOPT : 3691650.3

ZZ : 3655057.0

Iteratie 14:

ZR : 3013522.3

RES : 705565.8

PRIOPT: 3719088.0

DUOPT : 3719088.3

ZZ : 3678082.5

Iteratie 15:

ZR : 2878203.0

RES : 799879.4

PRIOPT: 3678082.5

Het in iteratie 15 bereikte optimum:

QW(1,J):	0.0	40000.0	0.0	0.0	0.0
QB(1,J):	0.0	0.0	40000.0	40000.0	40000.0
QX(1,K):	0.0	0.0	0.0	0.0	0.0
QW(2,J):	0.0	0.0	0.0	0.0	0.0
QB(2,J):	1000000.0	0.0	0.0	0.0	0.0
QX(2,K):	821879.4	211892.9	0.0	0.0	0.0
YS0(J) :	0.0	0.0	0.0	0.0	0.0
YS1(I) :	110000.0	211892.9			

TCG : 3678082.5

Testprobleem 2:

Listing van het parameterbestand PROGRAM.PAR

```
PARAMETER(CO=6,KU=5,M=5,N=2,SU=6,TU=25)
PARAMETER(NCP=N+M,ND=2*N+N*SU+N*KU)
PARAMETER(NTP=2*N*M+N+M+N*KU,NVP=N*M+N+M)
```

Invoerbestand voor het hulpprogramma MODIFY

```
1,85*0,
27*0,1,0,1,4*0,1,0,1,4*0,1,44*0,
7*0,5*1,0,5*1,9*0,2*1,15*0,3*1,39*0,
7*0,5*1,0,4*1,10*0,2*1,57*0,
26*0,2*1,0,1,4*0,1,0,1,4*0,1,0,1,4*0,2*1,0,2*1,33*0,
26*0,1,28*0,9*1,0,5*1,0,7*1,8*0,
43*0,1,4*0,2*1,0,2*1,33*0,
25*0,1,60*0,
17*0,1,7*0,1,18*0,3*1,39*0,
26*0,1,59*0,
 1,56,
42,44,
47,26,
17,18,
27,27,
36,27,26, 9, 7,16, 5,10, 7, 5,11,10,13,10,11, 9, 7,12, 7,15,
 7, 9,10,33,19,38,24,50,12,15,62,38,15,20, 9,29,23, 8,32,16,
34,23,45,11, 9,11,21,43, 9, 4,44,18,15,19, 9,12,11,11,10,11,
 8, 2,10, 7, 2, 4,11,10,12,10, 5,10,14,19,15, 6, 6,11,10,13,
11,10, 7, 9,12,11
```

Uitvoerbestand van het hulpprogramma MODIFY

```
22-10-1986
Testgebied: winputten 6 en 19, vraagpunten 1, 3, 7, 8 en 40
  6
100000
000011
011010
010010
000111
  36  92  53  81  50  70
  1  6  3  2  4
  0  1  1  2  3  3  1  2  1  2  3  4  5
   1
   1
   1   7
   1   4   7
   1   7   8
   1
   1   7
   1
   1   9
   1   9  10
```

```
   1   9  10  11
   1   9  10  11  12
  5
00101
00010
01000
11000
00100
  53  38  24  57 210
  5  4  2  1  3
  0  1  2  1  1  1  2  1  2  3  4  5
  14
  14  17
  14
  14
  14
  14  18
  14
  14  21
  14  21  22
  14  21  22  23
  14  21  22  23  24
```

ND: 25 SIMAX: 6 COMBMAX: 5

Invoerbestand voor het hoofdprogramma

```
22-10-1986
Testgebied: winputten 6 en 19, vraagpunten 1, 3, 7, 8 en 40
29100,100000,17000,32000,17.8,0.2080,5.0,50000.0,4.0,3.7,
 6,11,20,14,23,
56600,100000,17000,32000,19.9,0.4150,5.0,50000.0,4.0,1.6,
20, 7, 7,10, 4,
15000,350000,150000,1000000,40000
```

Uitvoerbestand van het hoofdprogramma

```
22-10-1986
Testgebied: winputten 6 en 19, vraagpunten 1, 3, 7, 8 en 40
```

Gebruikte parameters:

CO : 5 KU: 5 M : 5 N : 2 SU: 6 TU: 25

NCP: 7 ND: 25 NTP: 37 NVP: 17

Berekende variabelen:

TCG : 6749017.0

Iteratie 1:

ZR : 5930358.5

RES : 1313590.8

PRIOPT: 7243949.5

DUOPT : 7243950.0

ZZ : 5895737.5

Iteratie 2:

ZR : 9449961.0

RES : 288652.1

PRIOPT: 9738613.0

DUOPT : 9738612.0

ZZ : 6071909.01

Iteratie 3:

Het hoofdprogramma MAIN is gestopt omdat de startbasis van de subroutine RESIM niet toelaatbaar is