

## Enige in Nederland verkrijgbare personal-computer-softwarepakketten voor wiskundige programmering

**Citation for published version (APA):**

Rutten, W. G. M. M. (1989). Enige in Nederland verkrijgbare personal-computer-softwarepakketten voor wiskundige programmering. *Kwantitatieve Methoden*, 10(juni), 171-184.

**Document status and date:**

Gepubliceerd: 01/01/1989

**Document Version:**

Uitgevers PDF, ook bekend als Version of Record

**Please check the document version of this publication:**

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

[Link to publication](#)

**General rights**

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

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

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

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

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

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# ENIGE IN NEDERLAND VERKRIJGBARE PERSONAL-COMPUTER-SOFTWARE-PAKKETTEN VOOR MATHEMATISCHE PROGRAMMERING

Door: W.G.M.M. Rutten

## INLEIDING

---

Studenten in de Bedrijfskunde aan de Technische Universiteit Eindhoven kunnen een vak volgen, Mathematische Programmering, waarbij o.a. met behulp van een softwarepakket (SCICONIC) op een mainframe een drietal MP-problemen moet worden opgelost. Door de grote toename van het gebruik van PC's (ook bij studenten) rees de vraag of er geen pakket was geschikt voor op de PC. Aangezien er leveranciers zijn die een dergelijk pakket verkopen, werd een vijftal softwarepakketten voor mathematische programmering vergeleken. Dit artikel geeft een verkorte versie van het eindrapport. Het eindrapport [*Onderzoek naar 5 softwarepakketten voor mathematische programmering*] is verkrijgbaar bij de vakgroep Operational Research & Statistiek van de faculteit Bedrijfskunde aan de Technische Universiteit Eindhoven.

In hoofdstuk 1 zullen de criteria voor het vergelijkend onderzoek verduidelijkt worden. In de volgende vijf hoofdstukken worden de vijf geteste pakketten beschreven aan de hand van de opgestelde criteria. In hoofdstuk 7 staan de conclusies die uit dit onderzoek naar voren komen.

Het onderzoek heeft plaatsgevonden in de periode juli-augustus 1988 met de laatst beschikbare gegevens. Gezien de snelle ontwikkelingen op de software-markt is het mogelijk dat bepaalde opmerkingen/bevindingen in dit verslag reeds achterhaald zijn ten tijde van publicatie.

---

W.G.M.M. Rutten is een tweetal jaren werkzaam geweest op de Technische Universiteit Eindhoven, faculteit Bedrijfskunde o.a. als begeleider van het vak Mathematische Programmering.

Bereikbaar: DMV Campina bv, divisie Kunstmelkvoeders  
Postbus 441  
5460 AK Veghel  
tel. 04130-72067.

## **1. GEBRUIKTE TESTWIJZE**

---

Om de pakketten op een juiste manier te kunnen beoordelen, zijn er een aantal criteria (aandachtspunten) opgesteld waarmee de pakketten vergeleken kunnen worden. Hierbij is onderscheid gemaakt naar vier onderdelen: de handleiding, het pakket, extra mogelijkheden en de vereiste kennis. In de volgende hoofdstukken is de paragraafindeling gemaakt volgens deze vier onderdelen, in dit hoofdstuk wordt per onderdeel uitgelegd waarop gelet wordt.

### **1.1. De handleiding**

Een eerste blik wordt geworpen op de dikte van de handleiding. Het aantal pagina's is hier een maatstaf. Een tweede criterium is de leesbaarheid van de handleiding. Veel formules in de tekst heeft een negatieve invloed op de leesbaarheid; het gebruik van voorbeelden verbetert de leesbaarheid. Aanwezigheid van een lijst met mogelijke foutmeldingen is gemakkelijk voor de gebruiker. Verder zal van ieder pakket zeer beknopt een uitleg worden gegeven hoe de input eruit moet zien voor een correcte verwerking.

### **1.2. Het pakket**

Om de pakketten te kunnen vergelijken zijn er een aantal criteria opgesteld, die onderscheiden kunnen worden naar objectieve aandachtspunten en subjectieve aandachtspunten. De objectieve punten zijn:

- het aantal benodigde diskettes;
- de toegestane grootte van het model (aantal variabelen (continu, integer binair), aantal restricties en aantal niet-nul elementen in de matrix);
- de mogelijke codering van variabele- en restrictienamen;
- de rekentijd.

De subjectieve aandachtspunten zijn:

- de vereiste opzet van de input (benodigde tijd, overzichtelijkheid);
- de wijze van invoer (mogelijkheid fouten te verbeteren en/of veranderingen door te voeren);
- de output (beeldscherm, printer en/of file);
- de leesbaarheid van de output (objectwaarde, waarde van variabelen, restricties, restwaarden, marginale kosten);
- aanwezigheid van een rapportfunctie (bedieningsgemak).

### **1.3. Extra mogelijkheden**

In de derde paragraaf van de volgende hoofdstukken zal steeds ingegaan worden op eventueel aanwezige extra mogelijkheden van een pakket. Integer- en binaire variabelen zijn eigenlijk min of meer vereist, het gebruik van onder- en bovengrenzen aan variabelen (bounds) is meegenomen.

### **1.4. Vereiste kennis**

Verwacht wordt dat de gebruiker in ieder geval iets van DOS weet en zich moet inwerken in het pakket. Voor sommige pakketten is meer kennis nodig. Teveel inwerktijd is niet goed, omdat dat de aandacht van het eigenlijke onderwerp (mathematische programmering) afleidt.

### **1.5. Gevolgde werkmethode**

Per pakket werd de handleiding doorgenomen en beoordeeld. Vervolgens werden enkele simpele LP-problemen in het pakket ingevoerd en werd de input-wijze en de output beoordeeld. Een integer probleem werd toegepast om te testen of het pakket de afrondingsoplossing of de integer-oplossing geeft. Om dezelfde reden werd ook een binair probleem ingevoerd en door-

gerekend. Een Mixed-Integer-LP-probleem werd ingevoerd en ook een groter probleem (matrix  $\pm 100 \times 200$ ) werd getest.

Met deze werkmethode werd inzicht verworven in de werking en gebruiksvriendelijkheid van een pakket.

## 2. WHAT'S BEST

---

Het pakket What's Best is ontwikkeld voor spreadsheet gebruikers om hun "what if"-vragen te kunnen veranderen in "what's best"-vragen. Het is verkrijgbaar bij Computing & Systems Consultants in Eindhoven. Prijzen van het pakket zijn f 400,- (personal), f 1800,- (commercial) en f 2500,- (professional) exclusief BTW. Voor educatieve doeleinden worden kortingen geboden.

### 2.1. De handleiding

De handleiding is 98 bladzijden lang en engelstalig, doch zeer goed leesbaar door de vele voorbeelden die gebruikt worden. Over het algemeen zijn de voorbeelden gemakkelijk te noemen. Het hele programma draait om de spreadsheet Lotus 123. De handleiding van Lotus wordt echter niet gegeven. Het pakket What's Best bevat drie stappen:

1) het definiëren van "adjustable cells"

De "adjustable cells" zijn de variabelen die het programma een waarde mag geven (=beslissingsvariabelen van het MP-model). Let op: Iedere verwijzing naar één of meerdere adjustable cells telt ook als variabele!

2) het definiëren van "what's best"

De waarde van de objectfunctie moet in een bepaalde cel worden weergegeven, en die cel moet dan geoptimaliseerd worden. Het aangeven welke cel de objectwaarde bevat moet iedere keer opnieuw gedaan worden voordat men het What's Best programma start; dit wordt dus niet bewaard in de file.

3) het specificeren van restricties.

De restricties kunnen op twee manieren gespecificeerd worden. Ten eerste via de functietoetsen in What's Best. Men geeft aan dat de waarde van een bepaalde cel kleiner gelijk (" $<$ "), groter gelijk (" $>$ ") of gelijk (" $=$ ") moet zijn aan de waarde van een andere cel (rechterlid). Daarbij moet men één of twee kolommen achter het rechterlid vrijhouden voor de restwaarde(n). Ten tweede kan men gebruik maken van het feit dat geen enkele cel een negatieve waarde mag hebben in de oplossing. Met behulp van de restwaarde van een restrictie kan een cel met een negatieve waarde gecreëerd worden, die in de eindoplossing groter of gelijk aan nul zal zijn.

Voor foutmeldingen vanuit het pakket is een uitgebreide lijst met mogelijke foutmeldingen aan de handleiding toegevoegd.

### 2.2. Het pakket

Voor gebruik van het pakket heeft men twee diskettes nodig. De What's Best en een Lotus 123 diskette. De grootte van het pakket, in aantal variabelen, restricties e.d. varieert met de versie, zie tabel 2.1.

De codering van variabelen is niet mogelijk; het nummer van de cel duidt een variabele aan. Wel kan men in andere cellen tekst toevoegen, om de inhoud van een "variabele"-cel te verduidelijken, zie figuur 2.1.

Tabel 2.1. Specificaties van What's Best [handleiding What's Best, blz T-3]

	Versie: Personal	Commercial	Professional
<b>Aantal:</b>			
Numerieke cellen	800	3000	8000
Variabelen	250	1500	4000
Integer/binaire var.	25	150	400
Restricties	125	750	2000
Instructies	3000	18000	24000
Coëfficiënten	4000	24000	32000

Figuur 2.1. Gebruik van variabelen in What's Best.

	A	B	C	D	E	F	G
1							
2			prod A	prod B	totaal		max cap
3 Cap. fabriek I			0	0	0	<	330
4							

De cellen C3 en D3 zijn twee "adjustable cells". De som, C3+D3, staat in cel E3. Cel E3 moet kleiner of gelijk (cel F3) zijn aan 330 eenheden (cel G3). De namen van de variabelen zijn dus C3, D3 en E3. De formule (lees: restrictie) in cel E3 is alleen leesbaar als de cursor op cel E3 staat.

De opzet van de input is zeer fraai te maken doordat overal commentaar kan worden toegevoegd, zie figuur 2.1. Gebruikte formules zijn echter niet zichtbaar, dus controle op een correcte matrix is zeer moeilijk. Eventuele fouten zoeken kost dan ook zeer veel tijd.

De input kan men het beste op één van de volgende twee manieren opzetten:

- Tabelsgewijs

Men voert alle gegevens in de vorm van tabellen in en legt vervolgens tussen die tabellen alle nodige relaties. Voordeel is dat de input en output er zeer mooi uitzien, nadelen zijn dat bijna niet te controleren is of alle restricties al zijn ingevoerd en dat deze opzet veel extra tijd vergt.

- Matrixgewijs.

Hierbij voert men alle gegevens in de vorm van een enorme matrix in. Op de rijen staan de restricties en de kolommen stellen de verschillende variabelen voor. Deze manier is beter controleerbaar dan de vorige, mits men de matrix op fatsoenlijke wijze kan uitprinten. Bij grotere problemen is dit niet mogelijk, omdat het aantal karakters per printregel kleiner is als de breedte van de matrix. Het is minder mooi dan tabelgewijs.

De wijze van input is interactief in Lotus 123, dus fouten en dergelijke zijn direct te verhelpen, dat wil zeggen als ze ontdekt worden. De output is hetzelfde als de input, met op de plaats van de "adjustable cells" de waarde die de variabelen hebben aangenomen in de optimale situatie. Marginale kosten e.d. worden niet gegeven.

Het doorrekenen kost op een (IBM)PC-AT slechts een halve minuut voor een middelgrote opgave. De invloed van binaire variabelen op de rekentijd is verwaarloosbaar. Om een groter probleem (100\*200) fatsoenlijk te kunnen verwerken heeft men al gauw de commerciële versie nodig, omdat men anders zeker het maximum aantal variabelen overschrijdt; iedere verwijzing geldt

immers ook als variabele (voor dit onderzoek is de personal versie gebruikt).

De rekentijd hangt sterk af van de aanwezigheid van een coprocessor. Met coprocessor rekt de PC ongeveer drie keer zo snel (bovengenoemde PC heeft een coprocessor).

### 2.3. Extra mogelijkheden

Extra opties heeft What's Best niet. Zelfs integers zijn niet mogelijk. Wel kan men binaire variabelen opgeven. Bounds zijn alleen als restricties op te geven.

### 2.4. Vereiste kennis

Buiten het pakket What's Best is een redelijke kennis van Lotus 123 nodig. Mensen die niets van spreadsheets weten, zullen zéker een cursus in Lotus 123 nodig hebben voordat zij met What's Best uit de voeten kunnen.

## 3. NATHALIE

---

Het pakket Nathalie is ontwikkeld op de Technische Universiteit Eindhoven, faculteit Wiskunde. Het is ontstaan uit een mainframe pakket, dat geschikt is gemaakt voor de PC. Contactpersoon voor dit pakket is dhr. W. Keulemans. Het pakket is gemaakt in het kader van een onderzoek en wordt verder niet op de markt gebracht.

### 3.1. De handleiding

De handleiding van Nathalie bestaat uit ongeveer 4 pagina's. Zeer compact geschreven, dus ieder woord is belangrijk. Dit maakt de handleiding zeer moeilijk leesbaar. Bij Nathalie maakt de gebruiker met behulp van een editor een file waarin de probleembeschrijving staat. Het format van die file (procesformat) wordt in de handleiding uiteengezet. Vervolgens wordt het probleem via 4 programma's doorgerekend en wordt de oplossing geprint (zie schema 3.1).

Schema 3.1. Verwerking via 4 programma's bij Nathalie.

---

NEWPROC	LPINPUT	MIXINTB	NEWREPO
omzetten	-> omzetten	-> lp programma	-> rapporteert
proces in	shareinput		in procesvorm
shareformat	in geschikte		naar printer
	vorm		

---

De inputfile wordt als volgt opgezet:

- 1) Men definieert afzonderlijke processen binnen het probleem;
- 2) Per proces geeft men aan:
  - welke input van welk ander proces heeft dit proces
  - welke output naar welk ander proces heeft dit proces
  - wat zijn de relaties tussen de input en de output;
- 3) Bovendien geeft men aan welke kosten verbonden zijn aan welke stromen tussen twee processen.

### Voorbeeld 3.1. het procesformat.

---

```
PROCES  machine1
INPUT
  grondstofx  inkoop    x
  grondstofy  inkoop    y
  hulpstofa   afdeling1 a
OUTPUT
  produkt1   markt1    p11
  produkt1   markt2    p12
  produkt2   markt2    p22
RESTR
  0.34*x + 0.68*a - p12 - p22 = 0
  0.89*y - p11 = 0
  p11 + p12 + p22 <= 200
COST
  grondstofx  inkoop    machine1 -30
  hulpstofa   afdeling1 machine1 -20
  produkt1   machine1  markt2  50
```

---

De processen zijn: inkoop, afdeling1, machine1, markt1 en markt2. De onderscheiden stromen tussen de processen zijn:

- tussen inkoop en machine1: grondstofx en grondstofy
- tussen afdeling1 en machine1: hulpstofa
- tussen machine1 en markt1: produkt1
- tussen machine1 en markt2: produkt1 en produkt2.

De relaties tussen de input en de output worden gegeven m.b.v. de locale namen (x, y, etc.).

---

De naam van de objectfunctie is COST, maar de objectfunctie wordt standaard gemaximaliseerd. Als men wil minimaliseren in plaats van maximaliseren moet een regel in een programmafile veranderd worden (met een editor)! Dit is natuurlijk uit den boze!

Onder COST komen dus de opbrengsten van de verschillende stromen. Men geeft aan: de naam van de stroom, vanuit welk proces naar welk ander proces gaat de stroom en de opbrengst die daarbij hoort.

### 3.2. Het pakket

Voor het gebruik van Nathalie heeft men 1 diskette nodig, met daarop de 4 programmafiles. Voor het doorrekenen van een probleem moet de gemaakte file (in procesformat) op deze diskette staan, met de naam PROCINP.DAT.

De toegestane grootte van het model volgens de handleiding is: 200 rijen, 500 kolommen, maximaal 1200 elementen ongelijk nul en maximaal 20 integers.

Voor de naamgeving van processen en produktstromen heeft men 20 karakters ter beschikking, per proces geeft men iedere stroom ook een zogenaamde localname (maximaal 3 karakters, zie x, y, p11 e.d. in voorbeeld 3.1) die men gebruikt om de relaties weer te geven.

De proceswijze opzet van de input vergt wel een iets andere manier van denken, maar is zeer gemakkelijk om te gebruiken. Vooral bij grotere opgaven blijkt het voordeel zeer duidelijk.

Doordat de input met een editor gemaakt kan worden kunnen fouten gemakkelijk verbeterd worden. Door de modulaire opbouw van de input is het doorvoeren van veranderingen vrij gemakkelijk.

De uiteindelijke oplossing wordt direct door het programma (NEWREPO) geprint en kan niet eerst op het scherm gecontroleerd worden. Dit kost dus onnodig veel tijd en papier als men fouten maakt. Op de geprinte oplossing

zijn de waarden en de marginale kosten van variabelen en restricties direct af te lezen.

Overige opmerkingen:

Invoerfouten zijn te ontdekken, door de eerste stap van de verwerking (NEWPROC) te laten uitprinten. De foutmeldingen zijn echter niet altijd even duidelijk; een lijst met mogelijke foutmeldingen zou hier zeer verhelderend zijn.

Er bleek dat hoofd- en kleine letters niet uitwisselbaar zijn, dit in tegenstelling tot wat in de handleiding vermeld staat.

### 3.3. Extra mogelijkheden

Binnen Nathalie kunnen maximaal 20 integers gedefinieerd worden. Ook binaire variabelen zijn mogelijk. Het gebruik van bounds gaat via restricties. Let hierbij wel op, want een restrictie met slechts 1 variabele wordt gezien als bound, dus men mag de variabele geen coëfficiënt geven (zie voorbeeld 3.2). Dit is een fout in het programma.

Voorbeeld 3.2. bounddeclaratie

---

```
PROCES machine1
INPUT
  grondstofx   inkoop   x
RESTR
  3*x <= 1000
```

---

Bovenstaande restrictie op de hoeveelheid grondstofx die aangevoerd wordt, wordt verwerkt als zijnde een upperbound op grondstofx met een bovengrens van 1000, in plaats van 333,333!!!. Om de juiste bovengrens te krijgen had er het volgende moeten staan:  $x \leq 333.33333$ .

---

### 3.4. Vereiste kennis

Buiten de procesgewijze opzet van Nathalie moet men met een editor kunnen werken om de file aan te maken. Dit kan met behulp van EDLIN, of een willekeurige tekstverwerker. Aan de gebruiksvriendelijkheid, zoals andere (van oorsprong) PC-programma's die kennen, valt nog wel het een en ander te verbeteren.

## 4. PCPROG

---

Pcprog (versie 2.00) is een pakket voor LP, IP, MIP en QP problemen, ontwikkeld door E. Kalvelagen (Quantitative Management Software, Amsterdam). Het is bedoeld voor studiedoelinden en kleinschalige professionele modelbouw. Ook dit pakket is verkrijgbaar bij Computing & Systems Consultants te Eindhoven en het wordt in drie versies verkocht, te weten de personal-versie (f 350,-), midsize (f 1195,-) en professional (f 2500,-). Ook voor dit pakket gelden kortingen voor onderwijs.

### 4.1. De handleiding

De handleiding van Pcprog is 42 bladzijden lang en engelstalig. De handleiding is erg schematisch opgezet, maar wel te lezen. Pcprog heeft een eigen editor, maar men kan ook een andere gebruiken als men dat wenst. Het invoeren van het model kan op twee manieren:

1) Simple Parser: Men voert gewoon restrictie voor restrictie het model in (zie voorbeeld 4.1).



#### Voorbeeld 4.1. Input Simple Parser Pcprog

---

```
max 3x1+5x2
x1+4x2<=4
x1<=x2
```

---

- 2) Advanced Parser: Hierbij werkt men met een soort programmeertaaltje om zo de input compacter te maken (zie voorbeeld 4.2).

#### Voorbeeld 4.2. Input Advanced Parser Pcprog

---

```
$DECLARE x[1..5]
$SUM(i:=1 TO 5, 2x[i])<=5
```

---

Men moet variabelen eerst declareren (\$DECLARE).

De regel \$SUM... betekent:  $2x[1]+2x[2]+2x[3]+2x[4]+2x[5]<=5$ .

---

Er staan enkele kleine voorbeelden bij de uitleg van de verschillende mogelijkheden. Over foutmeldingen wordt niets gezegd.

### 4.2. Het pakket

Voor Pcprog heeft men 1 diskette nodig, waarop de programmafiles staan. Het programma is vrij kopieerbaar, dus is het aan te raden tijdens het werken een kopie van de programmafiles op de datadiskette te zetten. Ook kan met twee floppydrives of een hard-disk gewerkt worden.

Volgens de handleiding is de grootte van het geteste pakket 250 kolommen en 250 rijen. Alle 250 variabelen mogen integer zijn en 1500 elementen mogen ongelijk nul zijn.

De naam van variabelen mag tot 80 karakters lang zijn bij gebruik van de Simple Parser. Als men de Advanced Parser efficiënt wil gebruiken, moet men met variabelen met een teller (X1, X2, X3, etc) gaan werken. Hoewel de naamgeving dus vrijwel onbeperkt is, worden in de output alleen de eerste 12 karakters van iedere variabele gegeven. Hierdoor is het eigenlijke maximum aantal karakters 12.

De opzet van de input voor de Simple Parser vergt geen extra tijd, omdat die precies hetzelfde is als de manier waarop men het model opstelt. Voor de Advanced Parser is wel extra tijd nodig.

Doordat de input interactief is, kan men fouten gemakkelijk verbeteren. Als het model goed is opgezet is het doorvoeren van veranderingen ook makkelijk te doen.

Voor het doorrekenen van een middelgrote opgave was wederom ongeveer een halve minuut nodig. De grote opgave (100\*200) is niet gelukt (de inverse matrix, die tijdens de verwerking opgesteld wordt, werd te groot; in de zomer 1988 zou er een 1000x1000 versie op de markt komen die dit probleem niet meer zou hebben).

De output kan men zowel op het scherm als in een file krijgen. Die file kan men dan uitprinten. De output kan op drie niveaus:

- 1) Small: alleen de waarden van de objectfunctie en de variabelen worden gegeven.
- 2) Medium: ook het model wordt weergegeven.
- 3) Large: hetzelfde als medium én alle simplexiteraties worden gegeven (pivotelement, verdwijnend element, objectwaarde).

Een groot nadeel van de output is het gebruik van exponenten door het pakket. De objectwaarde is dan maar op 2 cijfers nauwkeurig af te lezen (bijvoorbeeld  $3,0E+06 = 3.000.000 \pm 49.999$ ).

Men kan een zogenaamde "Sensitivity Analyse" laten uitvoeren. Dan krijgt men de restwaarden, marginale kosten, een cost-coëfficiënt ranging en een RHS-rangings.

#### 4.3. Extra mogelijkheden

Binaire variabelen kan men via een omweg definiëren (een integer met een upperbound van 1). Verder heeft Pcprog geen extra opties.

#### 4.4. Vereiste kennis

Men moet alleen leren werken met Pcprog. Dit is gemakkelijk te leren omdat het pakket voor de gebruiker slechts bestaat uit een full-screen editor en functietoetsen. Verder is geen extra kennis vereist.

### 5. SAS/OR

---

Het programma SAS/OR (version 5) is een module van het pakket SAS. SAS is een software systeem voor data analyse. De OR-module is pas toegevoegd.

#### 5.1. De handleiding

In de User's Guide van SAS/OR wordt alleen de nieuwe module uitgelegd, niet hoe men moet starten met SAS (hiervoor moet men de "SAS User's Guide: Basics" doorworstelen). In de SAS/OR Guide (347 blz) moet men zo'n 75 bladzijden doorlezen over de LP-procedure. Er worden erg veel voorbeelden gebruikt.

De input voor SAS/OR gaat in de vorm van een coëfficiëntenmatrix (=data) die dan vervolgens met behulp van een (zelf geschreven) besturingsprogramma wordt doorgerekend. Zie voorbeeld 5.1.

Voor het doorrekenen van een probleem moet men een besturingsprogramma schrijven. Zie voorbeeld 5.2.

Door middel van de gegeven specificaties kan men de gewenste output produceren. In de handleiding worden dan ook 10 pagina's aan deze specificaties besteed.

#### Voorbeeld 5.1. Input SAS/OR

---

```
DATA PROBLEM;  
  INPUT X1 X2 X3 T $ R;  
  CARDS;  
1.5 2.1 3 MAX .  
1 -1 0 EQ 0  
1 1 1 LE 1  
;
```

---

Met DATA geeft men aan dat er gegevens komen. De INPUTkaart geeft aan dat iedere datakaart de volgende gegevens bevat; drie getallen (X1, X2 en X3), daarna volgt een "woord" (T \$) en als laatste gegeven weer een getal (R). Als een bepaald gegeven niet aanwezig is wordt dat op de betreffende kaart aangegeven met een "." (zie de eerste datakaart).

Bovenstaande input is het volgende probleem:

```
max! 1.5X1 + 2.1X2 + 3X3  
      X1 - X2 = 0  
      X1 + X2 + X3 ≤ 1  
      X1, X2, X3 ≥ 0
```

---

```
PROC LP DATA=PROBLEM;  
  VAR X1-X3;  
  RHS R;  
  TYPE T;
```

---

Met PROC LP roept men de procedure Lineaire Programmering aan. Daarbij wordt aangegeven dat de data die gebruikt wordt de naam PROBLEM (voorbeeld 5.1.) heeft. Daarna kan men allerlei specificaties geven. Hier zijn alleen VAR, RHS en TYPE opgegeven, men kan echter ook sensitivity/parametric/ranging control options vermelden.

---

## 5.2. Het pakket

Het aantal diskettes dat nodig is, is niet bekend omdat de geteste versie van SAS/OR op het VAXsysteem van de Technische Universiteit Eindhoven geïmplementeerd was en er geen PC-versie beschikbaar was (die zal volgen als het pakket voldoende beoordeeld wordt). De toegestane grootte van het model is nergens te vinden, maar gezien de wijze van input (coëfficiëntenmatrix) wordt de grootte in aantal variabelen beperkt door de lengte van een regel. Normaliter is een regel maximaal 80 karakters lang, waardoor het maximale aantal variabelen 40 bedraagt (tussen coëfficiënten moet een spatie zitten). Als men echter met coëfficiënten werkt die meerdere karakters lang zijn, wordt het aantal snel minder.

De codering van variabelen is beperkt tot maximaal 8 karakters. De opzet van de input vergt niet zo heel veel extra tijd, maar is wel erg onoverzichtelijk als men met grotere problemen gaat werken. De rekentijd op een PC is onbekend omdat het pakket op de VAX geïnstalleerd was.

De output kan op het beeldscherm verschijnen en ook geprint worden. De output is zeer volledig en duidelijk.

## 5.3. Extra mogelijkheden

Het pakket kent erg veel extra opties. Parametrisatie, RHS-ranging, sensitivity analyse van RHSsen en prijzen en natuurlijk ook integer variabelen zijn geen probleem. De vraag is echter of de PC-versie dit alles ook kan.

## 5.4. Vereiste kennis

Men zal iets van SAS moeten weten om een probleem te kunnen doorrekenen. Als men van de extra opties gebruik wil maken zal er echter meer studie van SAS nodig zijn.

Het gemakkelijkst is het maken van een batch-file, waarin alle commando's plus de data gegeven zijn, die men dan non-interactief door SAS laat verwerken.

## 6. OMP

---

OMP is een pakket van Belgische makelij (Beyers & Partners) en wordt in Nederland verkocht door Computing & Systems Consultants bv in Eindhoven. Prijzen van het pakket zijn f 275,- (tutorial), f 2500,- (midsize) en f 6950,- (professional) exclusief BTW.

### 6.1. De handleiding

De handleiding van OMP lijkt ontzettend dik (354 bladzijden), maar men kan starten met het pakket als er ongeveer 60 bladzijden doorgekeken zijn.

De handleiding is engelstalig, in de tekst staan veel voorbeelden ter verduidelijking. In de bovenbedoelde 60 bladzijden wordt één voorbeeld gehéél behandeld.

De input voor OMP kan op twee manieren geschieden:

1) explicit modeling mode

Hierbij voert men een model expliciet in. Met andere woorden, iedere restrictie wordt gegeven, zie voorbeeld 6.1.

Voorbeeld 6.1. Input in explicit modeling mode

---

```
MAXimaliseer algemeen probleem
X1 = PRODUCT1 = C $1
X2 = PRODUCT2 = C $1
*
C1 = MACHINE1 = PRODUCT1 + PRODUCT2 < 20
C2 = MACHINE2 = X1 + 2 X2 < 12
C3 = MACHINE3 = 2 X1 + 3 X2 < 30
C4 = MACHINE4 = 3 X1 + 2 X2 < 24
```

---

MAX betekent maximaliseer, rest van de regel is commentaar. Iedere variabele en restrictie krijgt een nummer (variabele Xi, constraint Ci). Vervolgens komt de "user defined name". Daarna komt bij de variabelen het type (Continu, Binair of Integer) en bij de restricties de restrictie (hierbij mag men zowel het variabelenummer als de -naam gebruiken). Als laatste wordt bij de variabelen nog de coëfficiënt in de objectrij gedefinieerd (na het \$-teken).

---

2) model generator mode.

Hierbij stelt men het model op en geeft via externe files (die ook via OMP aangemaakt kunnen worden) de getallen op. Het programma genereert dan het expliciete model. De model generator, voor het probleem dat in voorbeeld 6.1. is gegeven, staat in voorbeeld 6.2.

OMP beschikt over een editor, maar men kan ook met een eigen editor de files (ASCII) aanmaken. Het gehele programma is menu-gestuurd, waarbij het programma zelfs iedere keer de meest gebruikte keuze reeds invult zodat de gebruiker slechts op return hoeft te drukken.

In de handleiding is een lijst met mogelijke foutmeldingen aanwezig, met veel uitleg.

Voorbeeld 6.2. Input in model generator mode

---

```
MAX
SET = PRODUCT : PRODUCT1, PRODUCT2
SET = MACHINE : MACHINE1, MACHINE2, MACHINE3, MACHINE4
*
X = PRODUCT(&) = C $/COST/
C = MACHINE(&) = /COEF/*PRODUCT(S&) < /RHS/
*
DATA=COST, FILE=OBJECT, LINE=PRODUCT(&), COLUMN=COST
DATA=COEF, FILE=MATRIX, LINE=MACHINE(&), COLUMN=PRODUCT(&)
DATA=RHS, FILE=MATRIX, LINE=MACHINE(&), COLUMN=RHS
```

---

Men geeft wederom aan of het een maximalisatie of minimalisatie betreft, daarna definieert men de variabelen- en restrictienamen. Vervolgens geeft men de structuur van de variabelen en restricties aan en als laatste geeft men aan waar de data te halen zijn en hoe de data in de file zijn opgeslagen (rij en kolom). Het &-teken staat voor alle mogelijke waarden. Een externe waarde wordt aangegeven tussen slash's.

---

## 6.2. Het pakket

Het pakket bestaat uit twee diskettes (tutorial versie), die op een vaste schijf gekopieerd moeten worden voordat men kan werken. De grootte van het pakket in de verschillende versies wordt gegeven in tabel 6.1.

Tabel 6.1. Grootte van OMP

	Versie: Tutorial	Midsize	Professional
Aantal:			
variabelen	50	300	1000
restricties	20	150	500
integers	10	10	1000
elementen $\neq 0$	500	onbekend	3000

De codering van variabelen is vrij tot maximaal 24 karakters, hetzelfde geldt voor de restricties. De opzet van de input is in de explicit modeling mode vrijwel gelijk aan de gebruikelijke manier, alleen de definitie van de objectrij is afwijkend. Omdat de editor van OMP een full-screen editor is, kan men zeer gemakkelijk fouten verbeteren of veranderingen doorvoeren.

De rekentijd is wederom prima (ongeveer 2 minuten voor de grote opgave (100\*200), op een PC-AT).

Voor de output zijn er 10 standaard-rapporten aanwezig, die men echter naar believen kan aanpassen. Marginale kosten e.d. kunnen allemaal gegeven worden. De objectwaarde en de waarde van variabelen en restricties worden tot en met drie decimalen nauwkeurig gegeven. De output kan op het scherm, in een file of rechtstreeks geprint worden.

## 6.3. Extra mogelijkheden

Het gebruik van integer en binaire variabelen vormt voor OMP geen probleem. Ook kan men zonder problemen bounds definiëren, zie voorbeeld 6.3.

Er is een HELP-functie aanwezig, die vanaf elk menuscherm of vanuit de editor aangeroepen kan worden (via ?-teken). Nadat een optimale oplossing is verkregen kan men een user-simulatie toepassen. Dit houdt in dat men de waarde van variabelen en/of restricties kan veranderen, waarna het programma die nieuwe mogelijkheid doorrekent. Eventuele infeasibilites worden in de oplossing aangegeven.

Voorbeeld 6.3. Bounddeclaratie in OMP

```
MAX
X1 = PRODUCT1 = C $1 < 5
X2 = PRODUCT2 = C $1 > 2
```

De variabele X1 heeft een upperbound 5 en variabele X2 heeft een lowerbound van 2.

## 6.4. Vereiste kennis

Buiten het leren werken met de editor van OMP heeft men geen extra kennis nodig. Door de menu gewijze werking kan men zeer snel starten.

## 7. SAMENVATTING RESULTATEN & CONCLUSIE

---

In de tabel 7.1 zijn alle verzamelde gegevens samengevat. Men kan zien dat ieder pakket een aantal positieve en een aantal negatieve aspecten heeft. Voor de duidelijkheid worden hieronder de grootste voor- en nadelen per pakket gegeven:

- What's Best: - restrictie staat (onzichtbaar) in een cel  
+ output zeer fraai te maken
- Nathalie : - pakket is (nog) niet af  
+ procesformat zeer nuttig bij grotere modellen
- Pcprog : - grote opgave (100\*200) niet mogelijk  
- nauwkeurigheid objectwaarde laag  
+ zeer gemakkelijk te hanteren
- SAS/OR : - matrixgewijze opzet (alle nullen invoeren)  
+ zeer veel opties
- OMP : - moet geïnstalleerd worden op harde schijf  
+ zeer gemakkelijk te hanteren

De verschillen in rekentijd tussen de pakketten zijn verwaarloosbaar, de verschillen in rekennauwkeurigheid echter niet. Hierbij komt vooral Pcprog naar voren, in negatieve zin. Kijkt men naar de gebruiksvriendelijkheid dan springen Pcprog en OMP, die als stand-alone pakket zijn ontwikkeld, er ver bovenuit. What's Best en SAS/OR worden al snel complex en onoverzichtelijk en Nathalie is nog steeds een mainframepakket (dat op een PC draait).

Kortom, er zijn genoeg PC pakketten die beweren het gebied van wiskundige programmering te beheersen. De handleiding staat vol met prachtige voorbeelden, maar als men kritisch kijkt zijn er bij ieder pakket toch wel een aantal kanttekeningen te plaatsen die de gebruiksvriendelijkheid en toepasbaarheid verminderen met als gevolg dat het "geweldige pakket" niet geschikt blijkt voor bijvoorbeeld een cursus op een universiteit.

Tabel 7.1. Samenvatting verzamelde gegevens 1)

criterium	What's Best	Nathalie	Pcprog	SAS/OR	OMP
dikte handleiding (pp)	98	4	42	75	60
leesbaarheid handleiding	prima	moeilijk	formules	ok	prima
gebruik van voorbeelden	ok	redelijk	ok	veel	ok
foutmeldingen (handl)	ja	.	.	.	ja
aantal diskettes	2	1	1	1	2
grootte (# variabelen)	250	500	250	max 40	50
" (# restricties)	125	200	250	8	20
" (# elementen≠0)	800	1200	1500	?	500
" (# integers) 2)	0	20	250	40	10
" (# binaire var)	25	20	0	40	10
codering variabelen 3)	.	20	12	8	24
opzet input (tijd)	veel	redelijk	prima	redelijk	prima
opzet input (overzicht)	slecht	prima	ok	slecht	ok
foutmeldingen	ok	slecht	ok	prima	prima
doorvoeren veranderingen	moeizaam	ok	ok	moeizaam	ok
rekentijd	ok	ok	ok	?	ok
output (beeldscherm)	ok	.	ok	prima	prima
output (printer)	ok	ok	ok	prima	prima
leesbaarheid output					
" (waarde variabelen) 4)	vrij	2	max 5 5)	3	3
" (waarde restricties) 4)	vrij	2	max 5 5)	3	3
" (restwaarden) 4)	.	2	max 5 5)	3	3
" (marginale kosten) 4)	.	2	max 5 5)	3	3
" (objectwaarde) 4)	vrij	5	max 5 5)	0	3
gebruik van integers	0/1	ok	ok	ok	ok
gebruik van bounds	via ≤	via ≤	ok	ok	ok
extra kennis 6)	LOTUS	TVWK	.	SAS	.
prijzen					
personal	f 400,-	.	f 350,-	.	f 275,-
midsize	f 1800,-	.	f 1195,-	.	f 2500,-
professional	f 2500,-	.	f 2395,-	.	f 6950,-

- 1) De ingevulde scores van de pakketten op de criteria zijn gedeeltelijk subjectieve waarnemingen (zie hoofdstuk 1). Men moet ze gebruiken om de pakketten twee aan twee te vergelijken. Betekenis tekens: "?"= onbekend, "."= niet aanwezig, "ok"= okee.
- 2) Als bij integer en binair dezelfde waarde is ingevuld, betekent dit dat men of zoveel integers, of zoveel binaire, of een combinatie van die twee kan gebruiken.
- 3) Hier wordt bedoeld het maximale aantal karakters dat gebruikt kan worden.
- 4) Hier wordt het aantal cijfers achter de komma gegeven, dat elk pakket in zijn oplossing voor de verschillende items geeft. "vrij" wil zeggen dat de gebruiker zelf het gewenste aantal kan bepalen.
- 5) Dit varieert, bij grotere waarden gaat Pcprog met machten werken (bijvoorbeeld 2.5E+06). Het gegeven getal is het maximale aantal cijfers achter de komma dat Pcprog geeft.
- 6) Voor ieder pakket geldt dat de gebruiker DOS en het pakket moet kennen. Eventueel extra vereiste kennis wordt hier aangegeven. "TVWK" betekent TekstVerWerKer.