

Het gebruik van MGG en SCICONIC op de VAX t.b.v. het keuzevak "mathematische programmering" : handleiding voor studenten

Citation for published version (APA):

Verburg, J. (1985). *Het gebruik van MGG en SCICONIC op de VAX t.b.v. het keuzevak "mathematische programmering" : handleiding voor studenten*. (TH Eindhoven. THE/BDK/ORS, Vakgroep ORS : rapporten; Vol. 8507). Technische Hogeschool Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1985

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

Take down policy

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

openaccess@tue.nl

providing details and we will investigate your claim.

Het gebruik van MGG en SCICONIC op de VAX
t.b.v. het keuzevak "Mathematische Programmering"

handleiding voor studenten

Rapport ARW 03 THE BDK/ORS/85/07

Eindhoven, augustus 1985

J. Verburg

<u>Inhoudsopgave</u>	<u>pagina</u>
1. Inleiding	3
2. Volgorde van werken	4
3. SCICONIC	5
4. MPS invoer formaat	7
5. Matrix generatoren	12
6. VAX terminal sessie	14
7. Bestandsbeheer op de VAX	17
8. Eenvoudig gebruik van SCICONIC	21
9. Geavanceerd gebruik van SCICONIC	31
10. Gebruik van MGG	37
11. Fortran Conventies	50
12. Fouten en foutmeldingen	52

Bijlagen

- A MPS formaat
- B MGG invoer specificaties
- C MG invoer specificaties
- D SCICONIC integer and non-linear programming
- E Foutmeldingen

1. Inleiding

Bij het keuzevak "Mathematische Programmering" moeten oplossingen gevonden worden voor enkele aan de praktijk ontleende probleemstellingen. Hiertoe dient u modellen op te stellen die met behulp van de beschikbare software op de THE computers doorgerekend moeten worden.

Aan de TH Eindhoven is op DEC VAX computers standaard software beschikbaar voor mathematische programmering. De hier volgende beschrijving geldt voor de pakketten SCICONIC (Mathematische Programmering Oplossings Software) en MGG (Matrix Generator Generator) van Scicon Limited te Londen. Van beide pakketten is een Engelstalige User Guide beschikbaar.

Deze handleiding voor studenten is als eerste introductie bedoeld, voldoende voor eenvoudige situaties. Voor meer complexe situaties, formuleringen en uitwerkingen, zullen de genoemde User Guides geraadpleegd dienen te worden. De belangrijkste hoofdstukken uit de user guides zijn als bijlage hier bijgevoegd.

De lezer wordt geacht voldoende kennis van de theorie over modelvorming en oplossingsmethoden via de mathematische programmeringstechnieken te hebben. Zo nodig raadplege men de syllabus hierover: "Lineaire programmering langs pragmatische lijnen" van prof.dr. C.B. Tilanus, dictaatnummer 1.270.0.

2. Volgorde van werken

In het algemeen zal de hieronder aangegeven volgorde van werken aangehouden worden.

1. Probleemstelling bestuderen.
2. Model definiëren.
3. Gegevens verzamelen.
4. Invoer voor computerverwerking opstellen.
5. Computerterminal gebruiken om invoer te plegen en om een uitvoer t.a.v. de oplossing voor het probleem te produceren.
6. Uitvoer bestuderen en resultaten valideren.
7. Resultaten rapporteren.

In de praktijk zal het nodig blijken in voorkomende gevallen stappen terug te doen en stappen te herhalen.

In dit vervolg zullen we ons voornamelijk met de stappen 4 en 5 bezig houden.

3. SCICONIC

Op de VAX van de THE is het SCICONIC pakket voor mathematische programmering beschikbaar.

Dit programma heeft mogelijkheden voor:

- Lineaire Programmering (via de Simplex methode),
- Integer en niet-lineaire programmering (via de branch and bound techniek).

Bij dit laatste heeft het de mogelijkheden van:

- integer en 0-1 variabelen,
- semi-continue variabelen,
- special ordered sets.

Voor een uitvoeriger beschrijving van deze laatste mogelijkheden zie hoofdstuk 7 van de SCICONIC USER GUIDE.

3.1. Invoer

De gegevens-matrix moet in MPS formaat aangeleverd worden (eventueel m.b.v. een matrix generator gegenereerd).

Een beschrijving van het MPS formaat volgt hierna.

Een klein voorbeeldje:

```
NAME          SYLLAB30
*
* PRODUCTIE PROBLEEM UIT SYLLABUS LP (1.270.0) BLDZ 30
*
ROWS
  N  WINST
  L  MACHINE1
  L  MACHINE2
  L  MACHINE3
  L  MACHINE4
COLUMNS
  PRODX1  WINST      1.0
  PRODX1  MACHINE1   1.0          MACHINE2   1.0
  PRODX1  MACHINE3   2.0          MACHINE4   3.0
  PRODX2  WINST      1.0
  PRODX2  MACHINE1   4.0          MACHINE2   2.0
  PRODX2  MACHINE3   3.0          MACHINE4   2.0
```

RHS

UREN	MACHINE1	20.0	MACHINE2	12.0
UREN	MACHINE3	30.0	MACHINE4	24.0

ENDATA

3.2. Besturing

Besturing van SCICONIC gaat met behulp van kommando's die het mogelijk maken de vereiste verwerking aan de probleemsituatie aan te passen. Voor de verreweg meest voorkomende standaardsituaties zijn standaard commandosets beschikbaar gemaakt.

Een voorbeeld:

```
INFILE='MATRIX'      @ INVOER FILE
CONVERT              @ LEES DE INVOER
SETUP (MINIMISE)    @ KIES DE PROBLEEMSETTING
PRIMAL              @ SIMPLEX UITVOEREN
PRINTSOLN           @ GEEF DE RESULTATEN
OUTFILE='UITVOER'   @ NAAM VAN DE FILE VOOR DE UITVOER
PUNCHSOLN           @ PRODUCEER DE UITVOERFILE
STOP                @ EINDE RUN
```

3.3. Uitvoer

De resultaten kunnen op uw terminal getoond worden, maar ook op een bestand bewaard worden om uit te printen of te gebruiken voor latere verwerking. Eventueel kan dan een z.g. Report Generator gebruikt worden, zodat de resultaten in een voor leken begrijpelijke vorm en terminologie gepresenteerd kunnen worden. In deze handleiding zullen we de Report Generator overigens niet behandelen.

4. MPS invoer formaat

Het standaard MPS formaat is het de-facto standaard formaat geworden dat nu door nagenoeg alle L.P.-pakketten gebruikt wordt. Oorspronkelijk was het het invoerformaat voor het MPS (Mathematical Programming System) pakket dat op IBM 360 systemen beschikbaar was.

Naast de elementen van de gegevensmatrix kan via de MPS-formaat invoer o.a. ook de zg. BASIS gespecificeerd worden, hetgeen bij grote problemen nuttig en tijdsbesparend kan zijn.

In deze handleiding worden alleen de matrix gegevensspecificatie beschreven. Zie hoofdstuk 8 van de SCICONIC USER GUIDE voor een uitvoeriger beschrijving van het MPS formaat.

Traditioneel wordt een invoerregel nog vaak als een kaart (ponskaart) beschreven (posities = kolommen, max 80 posities, etc.).

De volgorde van de hier volgende specificaties is ook de volgorde waarin de verschillende gegevens ingevoerd dienen te worden.

4.1. NAME (verplicht aanwezig)

De eerste regel dient de NAME regel te zijn. Hiermee wordt een naam gegeven aan het probleem.

pos 1 t/m 4 NAME
pos 15 t/m 22 zelf te kiezen naam
Overige posities leeg laten.

4.2. * (commentaar, niet verplicht aanwezig)

Elke regel die * in positie 1 heeft wordt als commentaar beschouwd en verder nergens voor gebruikt. Commentaarregels mogen overal in de MPS invoer voorkomen.

4.3. ROWS (verplicht aanwezig)

Geeft aan dat onmiddellijk daarna de namen van de constraints (rijen) volgen.

pos 1 t/m 4 ROWS

4.4. Rijen gegevens:

pos 2	type v.d. constraint (N, E, L of G)
pos 5 t/m 12	zelf gekozen constraintnaam

De verschillende mogelijke rij-type codes zijn:

N = non restraint, d.w.z. de doelfunctie

E = equality, d.w.z. een gelijkheids constraint

L = less than or equal, d.w.z. kleiner of gelijk type

G = greater than or equal, d.w.z. groter of gelijk type

4.5. COLUMNS (verplicht aanwezig)

Geeft aan dat onmiddellijk daarna de namen van de kolommen (variabelen) en de matrix waarden volgen. Alleen de combinaties met waarden ongelijk aan nul hoeven te worden ingevoerd.

pos 1 t/m 7	COLUMNS
-------------	---------

4.6. Kolomgegevens (de eerste 3 velden verplicht aanwezig)

pos 5 t/m 12	variabelenaam
pos 15 t/m 22	constraintnaam
pos 25 t/m 36	matrix element waarde
pos 40 t/m 47	constraintnaam
pos 50 t/m 61	matrix element waarde

Alleen de waarden $\neq 0$ hoeven gegeven te worden. Alle gegevens behorend bij een en dezelfde matrixkolom dienen opeenvolgend opgegeven te worden.

4.7. RHS (verplicht aanwezig)

Geeft aan dat de daaropvolgende regels de RIGHT HAND SIDE (rechterdeel van de constraint vergelijkingen) bevatten. Alleen de waarden $\neq 0$ hoeven te worden ingevoerd.

pos 1 t/m 3	RHS
-------------	-----

4.8. Right hand side gegevens (De eerste 3 velden verplicht aanwezig)

pos 5 t/m 12	rechterdeel-naam
pos 15 t/m 22	constraintnaam
pos 25 t/m 36	waarde van het rechterdeel
pos 40 t/m 47	constraintnaam
pos 50 t/m 61	waarde van het rechterdeel

Vaak is er slechts één rechterdeel-naam in een probleem, maar het is mogelijk alternatieve situaties snel door te rekenen met meerdere right hand sides.

Bij de verwerking kan er i.h.a. slechts één tegelijk actief zijn.

4.9. RANGES (niet verplicht)

Geeft aan dat hierop volgend de ranges gegeven worden. Een range is een methode om twee constraints in één te rangen als het enige verschil in het rechterdeel is (b.v. ≥ 10 en ≤ 15 , geeft een range van 5).

pos 1 t/m 6	RANGES
-------------	--------

4.10. Ranges gegevens: (de eerste 3 velden zijn verplicht)

pos 5 t/m 12	rangenaam
pos 15 t/m 22	constraintnaam
pos 25 t/m 36	waarde van de range
pos 40 t/m 47	constraintnaam
pos 50 t/m 62	waarde van de range

Het effect van de rangewaarde is afhankelijk van het type van de rij.

Bij type E De waarde van de range wordt opgeteld bij de righthandside waarde. Als de waarde negatief is, is dus de som lager dan de righthandside waarde.

$RHS \leq RIJ\ WAARDE \leq RHS + RANGE$ of andersom
 $RHS + RANGE \leq RIJ\ WAARDE \leq RHS$ als de RANGE negatief is. (Rij-waarde betreft het linkerdeel van de constraint vergelijking.)

Bij type G De waarde van de range (die dan positief moet zijn) wordt opgeteld bij de righthandside waarde.

$$\text{RHS} \leq \text{RIJ WAARDE} \leq \text{RHS} + \text{RANGE}$$

Bij type L De waarde van de range (die positief moet zijn) wordt afgetrokken van de righthandside waarde.

$$\text{dus: } \text{RHS} - \text{RANGE} \leq \text{RIJ WAARDE} \leq \text{RHS}$$

4.11. BOUNDS (niet verplicht)

Geeft aan dat hierop volgend de bounds (begrenzungen) voor de variabelen (kolommen) volgen.

pos 1 t/m 6 BOUNDS

4.12. Bounds specificaties

pos 2 t/m 3 TYPE (UP, LO, FX, MI, PL, FR, UI, BV, SC)

pos 5 t/m 12 Bound naam

pos 15 t/m 22 variabelenaam

pos 25 t/m 36 waarde (niet bij MI, PL, FR of BV)

De mogelijke type codes zijn:

UP	bovengrens
LO	ondergrens
FX	vaste waarde, geen onderbegrenzing mag gelden
MI	ondergrens op min oneindig, hierbij geen waarde opgeven
PL	ondergrens 0, geen bovengrens, hierbij geen waarde opgeven
FR	geen begrenzungen aanwezig, hierbij geen waarde opgeven
UI	alleen gehele waarden mogelijk, bovengrens opgeven
BV	binaire (0-1) variabele, hierbij geen waarde opgeven

5. Matrix generatoren

Een Matrix generator is een programma om gegevens van de L.P. matrix in MPS formaat te produceren.

- De belangrijkste eisen aan een matrix generator te stellen zijn:
- ≠ Oorspronkelijk ruwe, niet reeds bewerkte gegevens (die herkenbaar zijn voor de gebruikers), moeten ingevoerd kunnen worden.
 - Modeldefinitie en gegevens manipulatie regels moeten onafhankelijk van de gegevens zelf gespecificeerd kunnen worden.

Het nut hiervan is dat:

- ⇒ Gegevensinvoer kan voorbereid en uitgevoerd worden door niet-specialisten (routine verwerking), ook bij veranderingen in de gegevens.
- Specialisten kunnen gemakkelijk wijzigingen aanbrengen als de situatie daarom vraagt (niet-routine zaken), bijvoorbeeld situaties die niet voorzien waren in de logica van het model.

MPS invoer kan in het algemeen op 3 manieren gecreëerd worden:

1. Met behulp van een z.g. editor, waarmee bestanden interactief aangemaakt of gemodificeerd kunnen worden. Dan moeten alle matrix elementen precies in de juiste positie ingetypt worden.
2. Met behulp van een klein programmaatje (Fortran, Algol, Basic, etc.) die zorgt dat de ingevoerde getallen precies in de juiste volgorde en kolommen neergezet worden. Zo'n programmaatje is voor u al gemaakt (MPSIN, te gebruiken via LPSOLVE).
3. Met behulp van een echte uitgebreide matrix generator, waardoor u niet de matrixwaarden zelf invoert, maar de ruwe gegevens waaruit door de matrix generator dan de matrix elementen berekend worden en in het goede formaat in een file gezet.

Voor kleine en eenvoudige problemen is een matrix generator niet nodig, voor grote, complexe en veranderende problemen is een matrix generator onontbeerlijk. Het gebruik is i.h.a. veel complexer dan van het LP oplossings-programma, omdat er veel meer te specificeren valt.

Beschikbaar op de VAX van de THE is o.a. MGG (Matrix Generator Generator) van dezelfde leverancier als SCICONIC.

Hierbij wordt de modelsituatie gespecificeerd waardoor een Fortran programma gegenereerd wordt om:

- invoer van de elementaire gegevens te plegen,
- volgens de modelspecificatie hieruit MPS formaat gegevens te produceren (de matrix),
- eventueel ook m.b.v. de L.P.-oplossing (uit SCICONIC) rapportage op maat te plegen.

6. VAX terminal sessie

Als u de beschikking heeft over een interactieve terminal (meestal een beeldschermterminal), dan moet u eerst inloggen (toegang krijgen) op het VAX systeem, d.w.z. u moet achtereenvolgens:

- # terminalapparatuur aanzetten,
- verbinding maken,
- identificatie opgeven.

6.1. Terminal aanzetten

De meeste beeldschermterminals worden aangezet door een schakelaar links aan de achterkant van het beeldscherm om te zetten.

6.2. Verbinding via het THENET

Voor terminals die niet rechtstreeks verbonden zijn met een bepaalde computer, maar via het THENET een keuze kunnen maken uit de verschillende computers, geldt de volgende procedure:

Alle computers hebben een bepaald adres; voor de VAX 11/750 computers (het VMS besturingsysteem) zijn deze:

Computer	THENET#adres
VAX-1/VMS	300
VAX#2/VMS	320

Als u op RETURN drukt krijgt u de THENET-prompt ("#"). U kunt nu een aantal commando's aan het THENET geven: HELP, DONE, CALL, ECHO en STATUS. Zie voor een uitvoeriger beschrijving van THENET het Rekencentrum Informatiebulletin AG#64.

Voor het werken met een VAX moet de locale echo (van het THENET) uitstaan. Hiervoor dient het commando ECHO OFF. Daarna maakt men met het CALL commando een verbinding met een computer: (de ingetoetste tekens zijn onderstreept)

```
# ECHO OFF  
# CALL 300
```

Als er een verbinding vrij is krijgt men de melding:

```
# CALL COMPLETED TO <adres>
```

Drukt men nu nogmaals op RETURN dan meldt de VAX zich.

6.3. Gebruikersidentificatie

De VAX-computer meldt zich met:

USERNAME:

Hierachter moet de aan u verschafte gebruikersnaam ingetypt worden, gevolgd door het indrukken van de RETURN-toets.

PASSWORD:

Nu moet het bijbehorende password ingetypt worden, ook weer afgesloten met de RETURN-toets.

Als het password wordt ingetypt verschijnen de letters niet op de terminal (no-echo) om te voorkomen dat iemand anders het password te weten komt.

Hebt u een juiste combinatie van gebruikersnaam en password correct ingetypt dan verschijnt op de terminal:

Welcome to VAX/VMS version 4.1

(hier staan verder eventueel berichten die voor de gebruikers belangrijk zijn)

\$

Tot slot staat op de eerste positie van de laatst gebruikte regel een \$-teken (de prompt) om aan te geven dat hierna een DCL (Digital Command Language) commando gegeven kan worden.

Wordt de gebruikersnaam of het password verkeerd ingetypt dan komt er een foutboodschap op de terminal.

Voorbeelden:

Username: GERARD

Password: (fout password ingetypt)

User authorisation failure

Username: GERARD

Password: (nu het goede password)

Welcome to VAX/VMS version 4.1

\$

6.4. Beeindigen

Wilt u de terminal-sessie beëindigen dan moet u dit kenbaar maken middels het commando:

\$ LOGOUT

Dus NIET door de terminal uit te zetten!

Het resultaat van dit commando is dat uw sessie beëindigd wordt; op de terminal verschijnt de gebruikersnaam en daarachter de datum en de tijd waarop dit commando is uitgevoerd.

LOGOUT/FULL geeft tevens informatie over o.a. de verbruikte CPU tijd en de "elapsed time", d.w.z. de tijd die verstreken is sinds het inloggen. Daarna komt de THENET prompt # op het scherm, en kan de apparatuur eventueel uitgeschakeld worden.

Voorbeeld:

```
$ LOGOUT
```

```
USER 42 logged out at 29-AUG-1985 15:17:15.50
```

Hierna volgt nog een melding van het THENET:

```
# SESSION 1 CLOSED TO 300,0
```

```
#
```

en u bent weer terug op het commando-niveau van het THENET

Meer gedetailleerde informatie over het gebruik van de VAX computer is te vinden in het rekencentrum informatiebulletin AG-75. Zorg dat u het gelezen hebt en dat u het zonodig kunt raadplegen tijdens uw gebruik van de terminals op de VAX.

6.5. Terminal eigenschappen

Afhankelijk van het type terminal, spelen bepaalde eigenschappen een rol. Bijvoorbeeld de breedte (aantal tekens) van een regel. I.h.a. hebben beeldscherm terminals regels van 80 posities, maar sommige kunnen meer (132) posities weergeven. Hiervoor kunt u dan soms gebruiken:

```
SET TERMINAL/WIDTH=132 en
```

```
SET TERMINAL/WIDTH=80
```

Ook bij beeldscherm terminals is er vaak een toets:

```
NO SCROLL (of BEELD VAST)
```

Hiermee kunt u de over het scherm lopende regels stopzetten of weer door laten gaan, teneinde de regels te lezen. Dit kan ook met de CTRL-S en CTRL-Q toets combinatie (voor stoppen en doorgaan).

Tenslotte is er vaak een interrupt toets, of break, om een run af te breken. In ieder geval is dit mogelijk met de combinatie CTRL-C (Control toets ingedrukt houden en C toets indrukken).

7. Bestandsbeheer op de VAX

Bij het gebruik van een computer, zoals de VAX, speelt het beheer van bestanden een grote rol bij het gebruik.

Om de verschillende voorkomende bestanden (gegevens, programma's, uitvoer, etc.) gemakkelijk te kunnen onderscheiden biedt de VAX een aantal mogelijkheden waarvan de belangrijkste hier vermeld worden.

Het begrip DEFAULT speelt bij computers een grote rol. Teneinde niet telkens weer onnodig veel dingen te moeten specificeren, worden er regels vastgesteld wat gebruikt wordt in het geval we bepaalde informatie niet geven. Letterlijk vertaald is "by default" in het Nederlands "bij verstek", dus bij afwezigheid.

7.1. Bestandsnamen

We identificeren elk bestand met behulp van een unieke naam die bestaat uit een aantal onderdelen:

```
[DIRECTORY.SUBDIRECTORY]FILENAAM.TYPE;VERSIE
```

Door het gebruik van goede defaults (dat is wat gebruikt wordt als we niets specificeren) zal in de praktijk vaak de volgende vorm gebruikt worden:

```
FILENAAM of FILENAAM.TYPE
```

Deze filenaam bestaat uit maximaal 9 alfanumerieke tekens (d.w.z. letters en cijfers, geen speciale tekens). Kies bij voorkeur een veelzeggende, makkelijk te onthouden naam:

```
MATRIXA
```

7.2. Type aanduiding van de bestanden

Deze zegt i.h.a. iets over de soort informatie die op de file te vinden is. De type codering bestaat i.h.a. uit 3 alfanumerieke tekens.

Enkele veelgebruikte type aanduidingen.

.DAT	data (gegevens)
.FOR	Fortran "source" programma tekst
.EXE	programma in direct verwerkbare vorm (executable)
.LIS	listing van een programma, uitvoer van de compiler
.DIR	directory (zie hieronder)
.TXT	tekst
.TMP	tijdelijk bestand
.OLB	programmabibliotheek
.COM	kommandobestand
.LOG	een zogenaamd LOG bestand

De default voor het TYPE van een bestand hangt van de situatie af. Niet altijd is er een default. Het TYPE code kan ook leeg zijn.

7.3. Versienummer

Dit is een volgnummer dat steeds opgehoogd wordt. Wanneer een bestand voor de eerste keer wordt aangemaakt krijgt het versienummer 1. Iedere keer als het gewijzigd wordt, wordt er een nieuwe file gemaakt waarvan het versienummer 1 hoger is. Het versienummer mag weggelaten worden, in dat geval geldt als default het hoogste nummer dat bij dit bestand voorkomt.

7.4. Directory en Sub*directory

De bestanden van een gebruiker kunnen door de gebruiker gegroepeerd worden in directories en subdirectories.

Onder een directory kunnen sub-directories vallen, waaronder op hun beurt weer sub-directories kunnen vallen, etc.

Bij specificatie voor de bestandsnaam worden ze dan gescheiden door een punt. DIRECTORY.SUBDIR.SUBDIR. etc.

Door een default directory te specificeren (met het commando SET DEFAULT) is het meestal niet nodig de directory bij de bestandsnaam op te geven.

Aanvankelijk heeft een nieuwe gebruiker slechts één directory voor zijn bestanden. De naam van deze directory bestaat uit (eventueel de eerste 9 posities van) zijn username. De default directory is daaraan gelijk.

Een sub-directory is te specificeren via het kommando:

```
$ CREATE/DIRECTORY      [USERNAME.TEST]
```

waarbij de subdirectory TEST toegevoegd wordt aan de directory USERNAME.

Doen we daarna \$DIR om te zien welke files er zijn in onze default directory [USERNAME], dan zien we o.a. de file TEST.DIR;1 als teken dat deze sub-directory bestaat.

M.b.v. SET DEFAULT [USERNAME.TEST] veranderen we de default directory, zodat als we nu \$DIR doen, we de inhoud van de sub-directory TEST zien. Aanvankelijk zal die leeg zijn.

Door voor elke nieuwe opdracht een nieuwe sub-directory te gebruiken kunnen we op een overzichtelijke, standaard manier werken, omdat in elke sub-directory zonder problemen dezelfde filenaam gebruikt kan worden (b.v. MATRIX.DAT).

Met \$SHOW DEFAULT kunt u op elk moment zien wat de default directory is.

7.5. Bestanden verwijderen

Met behulp van het commando \$ PURGE wordt van elke file waarvan meerdere versies bestaan, alleen die met het hoogste versienummer bewaard. De rest wordt verwijderd.

Via \$ DELETE FILENAAM.TYPE;VERSIE kan een specifieke file verwijderd worden. Het versienummer moet hierbij opgegeven worden.

Een .DIR file kan alleen verwijderd worden als de betrokken sub-directory leeg is.

7.6. Copiëren van bestanden

```
COPY FILE1.TYPE FILE2.TYPE
```

Copieert FILE1 naar FILE2.

Eventueel, als FILE2 al bestond, wordt er een nieuwe versie gemaakt. Zo kunnen ook bestanden van de ene sub-directory naar de andere gecopieerd worden.

Bijvoorbeeld, maak een nieuwe sub-directory en copieer een file MATRIX daar naar toe:

```
$ CREATE/DIRECTORY [USERNAME.PROB1]
```

```
$ COPY [USERNAME.TEST]MATRIX.DAT [USERNAME.PROB1]MATRIX.DAT
```

Maar i.p.v. deze laatste copy kunt u ook doen

```
$ SET DEFAULT [USERNAME.PROB1]
```

```
$ COPY [USERNAME.TEST]MATRIX.DAT MATRIX
```

8. Eenvoudig gebruik van Sciconic

Er is een (zo nodig conversationele) procedure LPSOLVE gemaakt om het gebruik van SCICONIC in eenvoudige gevallen heel gemakkelijk te maken. Voor de meer complexe gevallen is deze procedure niet geschikt.

LPSOLVE heeft vijf parameters die gelijk met de LPSOLVE opdracht gespecificeerd kunnen worden. Als ze niet direct meegegeven worden zal de procedure ze één voor één opvragen.

De vier parameters zijn (in deze volgorde):

1. Bestandsnaam van de invoermatrix in MPS formaat. Het default type is .DAT. Als de naam INPUT gebruikt wordt, zal middels een interactief programma (MPSIN), de MPS-matrix opgebouwd worden.
2. De bestandsnaam voor de resultaten van SCICONIC (de oplossing). Het default type is .DAT.
3. MAXIMIZE of MINIMIZE (MAX of MIN mag ook).
4. BATCH of INTERACTIEF (B of I mag ook). I is de default. Batch verwerking op een later (door de computer te bepalen) tijdstip gebeurt. We kunnen nog wel aangeven of heft overdag of 's nachts gebeuren moet ('s nachts gelden minder beperkingen dan overdag).
5. Continue of Integer oplossingsmethode (C of I). C is default. Een integer (of geheeltallig) probleem wordt opgelost (nadat het eerst als continue probleem opgelost is) via de Branch and Bound techniek. Dit kost i.h.a. zeer veel meer rekentijd.

Een voorbeeld:

```
LPSOLVE MATRIX UIT MAX INTERACTIEF
```

Hierdoor zal van het bestand MATRIX.DAT (van de op dat moment default directory) de MPS invoer gelezen worden. SCICONIC zal interactief uitgevoerd worden en een maximalisatie uitvoeren van een niet integer soort. De eventuele oplossing komt in de file UIT.DAT terecht om uit te laten printen of te bekijken. Via de interactieve verwerking zal op de terminal de verwerking gevolgd kunnen worden. (Gebruik zo nodig de NO SCROLL toets!.) Eventuele foutboodschappen verschijnen ook op de terminal. Schrijf ze op als dat gebeurt! De oplossing zal hierbij ook op de terminal verschijnen, maar na de run is dat verdwenen.

De file UIT.DAT kan op de terminal bekeken worden via:

```
$ TYPE   UIT.DAT
```

De breedte is echter 132 tekens, dus we doen eerst SET TERMINAL/WIDTH=132.

Uitprinten van de file gaat via:

```
$ PRINT   UIT.DAT           of  
$ PRINT/QUEUE=RC015  UIT.DAT
```

In de eerste vorm moet de uitvoer opgehaald worden uit de zogenaamde aflegvakken bij de computerzaal. In de tweede vorm wordt de (langzame) printer uit zaal RC15 gebruikt.

Bij BATCH verwerking (groepsgewijze, eventueel uitgestelde verwerking) wordt naast de resultatenfile ook een zogenaamde LOG file geproduceerd die de informatie die anders op de terminal zou komen bevat.

Deze file die de naam BATCH.LOG heeft, verschijnt na afloop van de batch job in uw directory.

Voorbeeld:

```
LPSOLVE  MATRixa  RESULT  MIN  BATCH
```

Leest van MATRixa.DAT, doet een minimalisatie m.b.v. SCICONIC in batch mode.

Het resultaat staat in RESULT.DAT en ook in LPMINB.LOG.

In deze laatste file staan ook eventuele foutmeldingen.

Gebruik dus \$ PRINT BATCH.LOG of \$ PRINT/QUEUE=RC015 BATCH.LOG. Doe dat vooral als er iets onverwachts gebeurt.

Een praktijkvoorbeeld van LPSOLVE volgt hierna. Merk op dat bepaalde fouten herkend worden, maar niet alle.

In veel gevallen zal een default gehanteerd worden als u als antwoord alleen de RETURN toets gebruikt.

In het voorbeeld zijn de ingetoetste tekens onderstreept.

In handschrift is commentaar toegevoegd.

QUESTION Wat is het inderbestand met de MATRIX in MPS formaat?

(Default is MATRIX.DAT)

Als u deze file nu eerst wilt maken type dan : INPUT

INPUT

Wie gaat de te maken file heten ? (default MATRIX.DAT)

MPSDATA Default type is .DAT dus MPSDATA.DAT

===== M P S FORMAAT INVOER GENERATOR THE =====

Geef een naam aan het LP model. (max. 8 tekens) (default is LPMODEL)
Niets ingetoetst, daarom de default naam gebruikt

LPMODEL

Geef de naam voor de DOEL-FUNCTIE (max 8 tekens) (default is OBJECT)

WINST

Geef de naam van een CONSTRAINT rij (max. 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

MACHINE1

Wat is het TYPE van de constraint (N, G, E of L)

L

Geef de naam van een CONSTRAINT rij (max. 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

MAXINE2

Deze tyfout kan niet ontdekt worden

Wat is het TYPE van de constraint (N, G, E of L)

K

Deze tyfout wordt wel ontdekt

Kies N(ND), G(GREATER THAN), E(EQUAL) of L(LESS THAN)

Wat is het TYPE van de constraint (N, G, E of L)

L

Geef de naam van een CONSTRAINT rij (max. 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

MACHINE3

Wat is het TYPE van de constraint (N, G, E of L)

!

Geef de naam van een CONSTRAINT rij (max. 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

MACHINE3

Deze constraint-naam is al seseven !

Geef de naam van een CONSTRAINT rij (max. 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

?

Hier door krijg ik de lyst van tot nu toe gebruikte

Bestaande constraint-namen en type :

constraint namen

WINST	N
MACHINE1	L
MAXINE2	L
MACHINE3	L

Geef de naam van een CONSTRAINT rij (max. 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

MACHINE4

Wat is het TYPE van de constraint (N, G, E of L)

L

Geef de naam van een CONSTRAINT rij (max. 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

Gereed met de constraints, dus niets ingetoetst, alleen RETURN

Geef de naam van een VARIABELE(kolom)(max 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

PROXY1

als alle namen seseven zijn dan alleen RETURN toets
PRODX2

Geef de naam van een VARIABELE(kolom)(max 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

? Nu krijg ik te zien welke variabelen er zijn
Bestaande variabele-namen :
PRODX1
PRODX2

Geef de naam van een VARIABELE(kolom)(max 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets
gereed met de variabelen

Geef de RIGHT HAND SIDE naam, of alleen RETURN (default is RHS)
UREN

Geef de RIGHT HAND SIDE naam, of alleen RETURN (default is RHS)
geen verdere right hand sides

Geef een RANGE naam, of alleen RETURN toets
geen ranges

Geef de naam van een BOUND, of alleen RETURN toets
geen bounds

Wilt u de matrix elementen ZELF OPGEVEN (JA)
of ze een voor een laten OPVRAGEN (NEE)
NEE Als de matrix groot en leeg is, is zelf opgeven te prefereren

Kies invoer volgorde, per CONSTRAINT of VAR (C/V)
C

Constraint : WINST
Geef de waarde bij variabele : PRODX1
1 getallen mogen als gehele waarden gegeven worden
Geef de waarde bij variabele : PRODX2
1.0 mag ook als reële waarde met decimaal punt

Constraint : MACHINE1
Geef de waarde bij variabele : PRODX1
1
Geef de waarde bij variabele : PRODX2
4
Geef de waarde in rhs : UREN
20

Constraint : MAXINE2
Geef de waarde bij variabele : PRODX1
1
Geef de waarde bij variabele : PRODX2
2
Geef de waarde in rhs : UREN
12

Constraint : MACHINE3
Geef de waarde bij variabele : PRODX1
2
Geef de waarde bij variabele : PRODX2
3
Geef de waarde in rhs : UREN
30

Constraint : MACHINE4
Geef de waarde bij variabele : PRODX1
3
Geef de waarde bij variabele : PRODX2
2
Geef de waarde in rhs : UREN
24

MATRIX STATISTIEKEN:

5 RIJEN

2 KOLOMMEN

10 NIET-NUL ELEMENTEN

100.00 PROCENT DICHTHEID

FORTRAN STOP

Wilt u de zojuist gemaakte matrix bekijken ? (J of N)

JA

```

NAME          LPMODEL
ROWS
N  WINST
L  MACHINE1
L  MAXINE2
L  MACHINE3
L  MACHINE4
COLUMNS
PRODX1  WINST      1.00000
PRODX1  MACHINE1   1.00000
PRODX1  MAXINE2    1.00000
PRODX1  MACHINE3   2.00000
PRODX1  MACHINE4   3.00000
PRODX2  WINST      1.00000
PRODX2  MACHINE1   4.00000
PRODX2  MAXINE2    2.00000
PRODX2  MACHINE3   3.00000
PRODX2  MACHINE4   2.00000
RHS
UREN    WINST      1.00000
UREN    MACHINE1   20.0000
UREN    MAXINE2    12.0000
UREN    MACHINE3   30.0000
UREN    MACHINE4   24.0000
ENDATA

```

Wilt u nu met SCICONIC de LP oplossins vinden? (J of N)

JA

Wat is de SCICONIC uitvoer file ?(Default RESULT.DAT)

UITVR

Specificieer MAXIMISE of MINIMISE svp (MAX of MIN)

MAX

Wilt u een INTERACTIEVE of een BATCH run ? (I of B)

het default antwoord is I hierop

Is dit een Continu of een Inteser run ? (C of I)

het default antwoord is C hierop

SCICONIC/VM VERSION VM/V1.30
 COPYRIGHT SCICON LTD. 1983

AUTHORISED FOR USE AT:
 UNIVERSITY OF TECHNOLOGY EINDHOVEN

Hier begint de
 interactieve uitvoer
 van SCICONIC .

1I>

1I>

ON ERROR GOTO ?

Als er fouten gevonden worden
 ga dan naar label 9

2I>

CONVERT
 PROBLEM LPMODEL - NEW VERSION

RHS VECTOR - UREN
 PROBLEM HAS 5 ROWS, 2 COLUMNS AND 10 NON-ZERO ELEMENTS
 CONVERT TOOK 0.56 SECONDS

Echter, niet echt inter-
 actief, want door de
 eerdere vragen zijn alle
 SCICONIC commando's al
 geyeneerd. U ziet wel
 wat er gebeurt .

```

3I>
SETUP(MAXIMISE)
PROBLEM LPMODEL ON FILE PROBLEM
CREATED ON 30-AUG-1985 09:36:14.23
PROBLEM HAS      5 ROWS,      2 COLUMNS AND      10 NON-ZERO ELEMENTS
RHS      - UREN
OBJECTIVE - WINST
INCORE MATRIX HAS      5 ROWS AND      2 COLUMNS
SETUP TOOK      0.86 SECONDS

```

```

4I>
PRIMAL

```

NITS	OBJECT	INFEAS	SECS
0	1.000000	0.000000(0)	1.02
2	-8.000000	0.000000(0)	1.09

SOLUTION IS OPTIMAL

```

5I>
PRINTSOLN

```

```

PROBLEM LPMODEL - SOLUTION NUMBER 1 - OPTIMAL
CREATED ON 30-AUG-1985 09:36:17.47 , AFTER 2 ITERATIONS
PRINTED ON 30-AUG-1985 09:36:18.04

```

```

...NAME...      ..ACTIVITY..  DEFINED AS
FUNCTIONAL      8.000000    WINST
RESTRAINTS

```

```

?:
?:

```

```

..ROW...  AT      ....ACTIVITY....
N  WINST    BS      -9.000000
L  MACHINE1 BS      18.000000
L  MAXINE2  UL      12.000000
L  MACHINE3 BS      21.000000
L  MACHINE4 UL      24.000000

```

```

*** END OF ROWS ***
?:

```

```

.COLUMN.  AT      ....ACTIVITY....
PRODX1    BS      6.000000
PRODX2    BS      3.000000

```

```

*** END OF COLUMNS ***
?:

```

```

6I>
PUNCH=0

```

```

7I>
9. CONTINUE

```

```

8I>
STOP
FORTRAN STOP

```

*Nu in de verwerking
beëindigd*

LPMODEL

PROBLEM LPMODEL - SOLUTION NUMBER 1 - OPTIMAL
 CREATED ON 22-AUG-1985 15:50:56.98 , AFTER 2 ITERATIONS
 PRINTED ON 22-AUG-1985 15:51:01.15

...NAME... ..ACTIVITY.. DEFINED AS
 FUNCTIONAL 8.000000 WINST
 RESTRAINTS UREN

LPMODEL

NUMBER	..ROW...	ATACTIVITY....	.SLACK	ACTIVITY.	..LOWER BOUND...	..UPPER BOUND...	.DUAL ACTIVITY..
N	1	WINST	BS	-9.000000	8.000000	NONE	NONE	1.000000
L	2	MACHINE1	BS	18.000000	2.000000	NONE	20.000000	.
L	3	MACHINE2	UL	12.000000	.	NONE	12.000000	0.250000
L	4	MACHINE3	BS	21.000000	9.000000	NONE	30.000000	.
L	5	MACHINE4	UL	24.000000	.	NONE	24.000000	0.250000

LPMODEL

NUMBER	.COLUMN.	ATACTIVITY....	...INPUT COST...	..LOWER BOUND...	..UPPER BOUND...	..REDUCED COST..
	6	PRODX1	BS	5.000000	-1.000000	.	NONE
	7	PRODX2	BS	3.000000	-1.000000	.	NONE

\$

- 27 -

... moet de BATCH run in de DAG of in de NACHT ...

... CONTROL (ogeele DAY, entry 409) started on VARS011801
DIE BATCH.LOG

Kijk of de LOG file al gemaakt is.
Als de grootte op 0 staat in hij nog niet gereed met de batch job.

... directory USER2:EDDOR2

14 26-AUG-1985 11:46

Total of 1 file, 14 blocks.

TYPE BATCH.LOG

Bekijk de LOG file om te zien of alles goed ging

```

$ ! System Login Command Procedure
START:
ON CONTROLLY THEN GOTO START ! Don't allow control/s
SET NOON
! set default file protection
SET PROTECTION=(SY:RWED,OW:RWED,GR:RWED,WO:RE)/DEFAULT
! enable control/T handling
SET CONTROL=T
! listen to all broadcast-messages
IF F#MODE() .EQS. 'INTERACTIVE' THEN SET BROADCAST=ALL
! define foreign commands
US*ERS      ::= SHOW USERS
TI*ME      ::= SHOW TIME
NEWS       ::= TYPE SYS#NEWS:NEWS.TXT
! kermit defines
KERMIT     ::= $SYS$SYSTEM:KERMIT.EXE
! SDRG DEFINES:
IDEAS     ::= @SDRG_CAE:SDRDCAE.COM
! MARC DEFINES:
MARC      ::= @USER3:[MARC1]MARC
UMARC     ::= @USER3:[MARC1]UMARC
FT        ::= @USER3:[CWHANSL.PROGRAMS.MARC]BTJFT
! SAS DEFINES:
SAS       ::= $SYS$SYSROOT:[SAS410.IMAGE]SAS410.EXE
! MATLAB DEFINES:
MATLAB    ::= RUN MATLAB
! SPSSX:
SPSSX     ::= $SPSSX
! MAPPS:
MAPPS     ::= RUN USER2:[MAPPS]FRONT
! SOS:
SOS       ::= $SOS
! Vanweese een Fortran compilerfout: (d.d. 19-aug-1985)
F#ORTRAN  ::= FORTRAN/NOOPTIMIZE
! message of the day
TYPE SYS#RCCOMMON:MOTD.TXT

```

Dit is de verwerking van de Login Command procedure en dit is voor uw doeleinden niet van belang.

Zoek naar het begin van de Sciconic uitvoer



Type 'NEWS' voor nieuws d.d. 22-AUG-1985

\$ EXIT

```

$ ! sample user login command procedure
$ ! 7.AUG.84
SET NOON
SET PROTECTION=(SY:RWED,OW:RWED,GR:RE,WO)/DEFAULT
! useful abbreviations
DIR*ECTORY ::= DIRECTORY/DATE/SIZE
SHD        ::= SHOW DEFAULT
UP         ::= SET DEFAULT [-]
OP         ::= SET DEFAULT [-]
TOP        ::= SET DEFAULT SYS$LOGIN
SQ         ::= SHOW QUEUE/BATCH/ALL/DEVICE
H*HOME     ::= SET DEFAULT SYS$LOGIN
SYS        ::= SHOW SYSTEM
LO         ::= LOGOUT/FULL
BREED     ::= SET TERMINAL/WIDTH=132
SMAL       ::= SET TERMINAL/WIDTH=80
DOWN      ::= @CBDR2]NEER
NEEP      ::= @CBDR2]NEEP
NAAR      ::= @CBDR2]NAAR
! set .LOG for all commands
DEL        ::= DELETE/LOG
LIB        ::= LIBRARY/LOG
PUR        ::= PURGE/LOG
REN        ::= RENAME/LOG
! Inquire terminal-type
SET TERMINAL/INQUIRE

```

```

GET-W-NOTSET1- error modifying HSC000#DUA1:
DLI-E-INVALIDTYPE, invalid device type - specify a mailbox device
) MATH.PROG.
ASSIGN USER2:CLP]HELFILE.DAT HELPLINE
LP := $USER2:CLP]SCICONIC.EXE
LPSOLVE := @USER2:CBDR2]LPSOLVE
@USER2:CLP.MGG]MGGDEF

```

SYMBOL DEFINITIONS FOR MGG

```

ASSIGN user2:[!f.mss] MGG$DIR
MGG:==*@MGG$DIR:MGG/OUTPUT=SYS$OUTPUT $$$*
MGCL:==*@MGG$DIR:MGCL.COM*
RWL:==*@MGG$DIR:RWL.COM*
MG:==*@MGG$DIR:MG/OUTPUT=SYS$OUTPUT $$$*
RW:==*@MGG$DIR:RW/OUTPUT=SYS$OUTPUT $$$*

```

```

EXIT
EXIT
ET DEFAULT [BDOR2]
P -I MATRIX -O RESULT -P PROBLEM

```

SCICONIC/VM VERSION VM/V1.30
COPYRIGHT SCICON LTD. 1983

*Hier begint de
SCICONIC uitvoer*

AUTHORISED FOR USE AT:
UNIVERSITY OF TECHNOLOGY EINDHOVEN

```

ID
CONVERT
CONVERT
PROBLEM LPMODEL - NEW VERSION
RHS VECTOR - UREN
PROBLEM HAS 5 ROWS, 2 COLUMNS AND 10 NON-ZERO ELEMENTS
CONVERT TOOK 0.49 SECONDS

```

```

SI
SETUP(MAXIMIZE)
SETUP(MAXIMIZE)
PROBLEM LPMODEL ON FILE PROBLEM
CREATED ON 26-AUG-1985 11:48:42.31
PROBLEM HAS 5 ROWS, 2 COLUMNS AND 10 NON-ZERO ELEMENTS
RHS - UREN
OBJECTIVE - WINST
INCORE MATRIX HAS 5 ROWS AND 2 COLUMNS
SETUP TOOK 0.76 SECONDS

```

```

SI
ON ERROR GOTO 9
ON ERROR GOTO 9

```

```

SI
PRIMAL
PRIMAL

```

NITS	OBJECT	INFEAS		SECS
0	1.000000	0.000000	0)	0.89
2	-3.000000	0.000000	0)	0.95

SOLUTION IS OPTIMAL

```

SI
PRINTSOLN
PRINTSOLN

```

PROBLEM LFMODEL - SOLUTION NUMBER 1 - OPTIMAL
 CREATED ON 26-AUG-1985 11:46:44.53 , AFTER 2 ITERATIONS
 PRINTED ON 26-AUG-1985 11:46:44.77

...NAME... ..ACTIVITY... DEFINED AS
 FUNCTIONAL 8.000000 WINST
 RESTRAINTS UREN

LFMODEL

NUMBER	..ROW...	ATACTIVITY....	.SLACK	ACTIVITY.	..LOWER BOUND...	..UPPER BOUND...	.DUAL ACTIVITY..
N	1	WINST	BS	-9.000000	8.000000	NONE	NONE	1.000000
L	2	MACHINE1	BS	18.000000	2.000000	NONE	20.000000	.
L	3	MACHINE2	UL	12.000000	.	NONE	12.000000	0.250000
L	4	MACHINE3	BS	21.000000	9.000000	NONE	30.000000	.
L	5	MACHINE4	UL	24.000000	.	NONE	24.000000	0.250000

LFMODEL

NUMBER	.COLUMN.	ATACTIVITY....	...INPUT COST...	..LOWER BOUND...	..UPPER BOUND...	..REDUCED COST..
6	PRODX1	BS	6.000000	-1.000000	.	NONE	.
7	PRODX2	BS	3.000000	-1.000000	.	NONE	.

6I:
 PUNCHSOLN
 PUNCHSOLN

7I:
 Y CONTINUE
 P CONTINUE

8I:
 STOP
 STOP

FORTRAN STOP
 DDOR2 Job terminated at 26-AUG-1985 11:46:51.14

Accounting information:
 Buffered I/O count: 74 Peak working set size: 1244
 Direct I/O count: 106 Peak page file size: 16742
 Page faults: 1218 Mounted volumes: 0
 Charged CPU time: 0 00:00:10.26 Elapsed time: 0 00:00:18.61

Einde van de Batch job

\$

- 30 -

9. Geavanceerd gebruik van SCICONIC

Als de standaard besturingscommando's voor SCICONIC die door LPSOLVE gebruikt worden (zie het voorbeeld daarvan uit hoofdstuk 3) niet voldoen, moet een eigen reeks besturingscommando's opgegeven worden. Dit kan zowel interactief als in batch mode gebeuren. We moeten dan wel meer weten over SCICONIC.

9.1. Bestanden

Bestanden die door SCICONIC gebruikt kunnen worden:

- * de invoerfile met de MPS-formaat matrix;
- de zogenaamde PROBLEM file, waarin door SCICONIC op efficiënte manier de probleemsituatie wordt bewaard en van waar de situatie weer ingelezen kan worden. Als geen naam gegeven wordt, wordt een "temporary" file gebruikt;
- de SOLUTION file die de oplossingsresultaten bevat op een zodanige manier dat de Report Writer van MGG daarmee werken kan;
- de CONTROL file

Dit is niet echt een bestand maar de besturingscommando's in het geheugen van SCICONIC. Deze commando's zijn wel met het commando PUNCHCONTROL op de uitvoerfile te zetten en met het @ADD besturingscommando zijn ze weer te laden en uit te voeren;

- de RANGE file wordt beschreven door het RANGING commando en kan m.b.v. het PRINTRANGE commando uitgeprint worden.

9.2. Aanroep

De algemene opdracht om SCICONIC te gebruiken is:

LP

Hierbij kunnen een aantal parameters gegeven worden (de meeste in groepen van twee), gescheiden door één of meer spaties.

- *I gevolgd door de naam van de invoerfile
- O gevolgd door de naam van de uitvoerfile
- *S gevolgd door de naam van de SOLUTION file
- R gevolgd door de naam van de RANGE file
- P gevolgd door de naam van de PROBLEM file

Bovendien kan nog +B of -I (één van beide) gegeven worden om batch of interactieve verwerking aan te geven. Default gebruik via de terminal is interactief.

Voorbeeld: LP -I MATRIX -O UIT1

9.3. Besturing

Na deze LP opdracht verwacht SCICONIC de besturingscommando's te krijgen. In SCICONIC terminologie is dit de CONTROL AGENDA. Hoofdstuk 5 uit de SCICONIC user guide geeft een volledige opsomming. De belangrijkste volgen hier.

Bij de meeste van deze opdrachten spelen bepaalde waarden of namen een rol die in zogenaamde SSV's (System State Variables) bewaard worden.

b.v. CONVERT leest invoer van de file waarvan de naam in de SSV INFILE is opgeslagen.

Zo'n SSV kan een waarde gegeven worden door

SSVNAAM=waarde

Als de waarde een tekst is moet ze tussen aanhalingstekens ('') staan; bijvoorbeeld

INFILE='MATRIX'

Ook eigen benodigde variabelen (user variables) mogen gedefinieerd en gebruikt worden.

Deze kunnen slechts één van de volgende types zijn:

Double Precision	aan te geven met	DP
Real	aan te geven met	REAL
Integer	aan te geven met	INT
Logical (TRUE of FALSE)	aan te geven met	LOG
Hollerith (tekst)	aan te geven met	HOL

De belangrijkste besturingscommando's:

- HELP Geeft informatie over besturingscommando's en SSV's. Het is zeer aan te bevelen dit eens te bekijken.
- GOTO nummer Sprongopdracht om het besturingscommando waar dat nummer voor gezet is, nu uit te gaan voeren.
- ON ERROR..... Specificeert wat te doen bij het detecteren van fouten tijdens de run (b.v. ON ERROR GOTO 90).
- ECHO en BRIEF ECHO houdt in: geef elk commando eerst weer op de terminal. BRIEF is het tegenovergestelde van ECHO.
- @ADD filenaam Lees verdere commando's of input vanaf de genoemde file. Als het einde van die file bereikt is, lees dan weer van de terminal.
- CHECKON en CHECKOFF Voer de daaropvolgende commando's niet uit (CHECKON) maar controleer ze wel. CHECKOFF zet de CHECKON situatie weer af.
- CONVERT Leest de MPS matrix van de invoerfile die in de SSV INFILE genoemd wordt en converteert naar het interne formaat in de PROBLEM file. Als daarvoor geen filenaam was opgegeven wordt een tijdelijke hulpfile gebruikt.
- REVISE Wijzigt een matrix van de PROBLEM file. Invoer van de wijzigingen komen vanaf de terminal of via @ADD filenaam vanaf een bestaande file. Revision input heeft een op MPS gelijkend formaat met daarin MODIFY, DELETE, BEFORE ... en AFTER ... specificaties.
- SETUP(.....) Leest de PROBLEM file en laad het gekozen probleem in het geheugen. Als parameter kunnen o.a. MAXIMISE en MINIMISE meegegeven worden tussen

haakjes. MAXIMIZE heeft tot gevolg dat de SSV SCALE de waarde $\neq 1$ krijgt. SCICONIC zal altijd een minimalisatie doen, de cost coëfficiënten veranderen alleen van tekens. Dat ziet u ook aan de resultatenuitvoer. Default is minimalisatie. In dat geval is SETUP voldoende, zonder parameters.

PRIMAL of DUAL . Start het simplex algoritme dat gebruikt wordt om het minimum te vinden. Een iteratieverslag komt op de terminal. Normaal wordt PRIMAL gebruikt.

PRINTSOLN Produceert de resultaten op de terminal. Bij interactieve verwerking kunt u gedetailleerd specificeren hoe. Als u geen schrijvende terminal gebruikt zijn de resultaten na afloop verdwenen. Gebruik dan PUNCHSOLN.

PUNCHSOLN Produceert een bestand met de oplossing voor de printer op de file die door de SSV OUTFILE aangegeven wordt. De regelbreedte is 132 posities.

GLOBAL Start het branch en bound algoritme om vanuit een oplossing van het continue probleem, nu het geheel-tallige probleem op te lossen. Zie voor meer details de user guide.

RANGING Post optimum analyse. Hiermee wordt bepaald bij welke verandering in de kost de basis gaat veranderen; de kost konsekwenties als een activiteit niet op de optimale waarde is en de konsekwentie van het veranderen van een constraint (en binnen welke grenzen dit geldt).

PRINTRANGE en PUNCHRANGE Produceert de resultaten van de RANGING op de terminal (PRINTRANGE) of in de OUTPUT file (PUNCHRANGE).

PARACOST Post optimale parametrisatie die de optimale oplossing volgt bij verandering van de cost coëfficiënten. De verandering in de object functie wordt bepaald door een zogenaamd change object functie, die als een N type constraint in het probleem gedefinieerd moet zijn en aangewezen door de SSV OBJ CHANGE voor SETUP wordt uitgevoerd. De SSV CURPHI bepaald hoeveel keer de waarde van de cost coëfficiënt uit de change object functie per parametrisatiestap opgeteld worden bij de originele cost coëfficiënten. Dus: Object = Orig.Object + CURPHI * Change Object.

PARARHS Doet vergelijkbare postoptimale parametrisatie t.a.v. de RHS. De zogenaamde Change RHS wordt aangegeven via de SSV CHANGE en de multiplier in de SSV CURTHETA. Dus RHS = Orig RHS + CURTHETA * Change RHS.

PUNCHMATRIX Produceert een MPS formaat matrix op de OUTPUT file. Kan nuttig zijn na een REVISE operatie.

PUNCHCONTROL Zet de interne CONTROL file op de OUTPUT file, zodat later via @ADD deze control file weer gebruikt kan worden.

STOP Einde van de SCICONIC verwerking.

Complexer voorbeeld: (commentaar na een @ teken)

LP

INT J @ definieert J als een integer

J=0 @ zet de waarde van J op 0

INFILE='MIJNMAT' @ MPS invoer

PROBFILE='MIJNPROB' @ PROBLEM file blijft nu bewaard

CONVERT @ lees invoer naam problem file

SETUP @ laad de gegevens in het geheugen. Het is een minimalisatie

PRIMAL @ zoek optimum voor continue probleem

PRINTSOLN @ laat oplossing zien

```
IF(FEASIBLE.EQ.FALSE)GOTO 20 @ conditie op basis van SSV FEASIBLE
10 GLOBAL (USER) @ Integer optimalisatie. User bepaalt
zelf wat te doen als er iets bijzonder
is.
PRINTSOLN @ laat deze integer oplossing zien
J = J+1 @ J is een teller voor de hoeveelheid
integer oplossingen gevonden zover
IF (J.LT.5) GOTO 10 @ we willen maximaal 5 geheeltallige
oplossingen zien
20 STOP @ einde run.
```

10. Gebruik van MGG

10.1. Algemeen

De MGG software dient, om ook in complexe situaties, de invoer voor SCICONIC te maken, uitgaande van de ruwe (onbewerkte) gegevens van de probleemstelling.

Omdat het aantal verschillende mogelijkheden hierbij zeer groot is, zal dit niet gaan via een interactief, vraag en antwoord programma. We gebruiken hier een z.g. specificatietaal; dit vormt de invoer voor MGG.

MGG is een Matrix Generator Generator, d.w.z. volgens de gegeven specificaties wordt door het MGG programma een Fortran programma opgebouwd dat op zijn beurt de Matrix Generator gaat worden. Deze Matrix Generator (het MG programma) leest op zijn beurt de ruwe gegevens en bouwt de MPS-formaat file hier uit op.

De volgorde van werken is dus:

- Opstellen van het LP model en vaststellen welke ruwe gegevens er zijn.
- Invoer specificaties voor MGG geven, waarmee de regels vastgelegd worden hoe de LP matrix opgebouwd moet worden.
- Invoer voor MG geven, d.w.z. ruwe gegevens invoeren.
- Sciconic verwerking doen, d.w.z. de LP oplossing laten berekenen.

Additionele stappen zouden kunnen zijn in het kader van de Report Writer die bij MGG hoort:

- Opstellen van de rapport specificaties
 - Draaien van de report module op basis van de Sciconic resultaten.
- Deze twee stappen worden echter niet van u gevraagd.

10.1. Voorbeeld

De MGG invoer specificaties worden beschreven aan de hand van een klein voorbeeldje uit de MGG User Guide.

In bijlage ... vindt u de volledige beschrijving van de mogelijke MGG invoer specificaties.

Het gekozen voorbeeld is een mengprobleem:

$$\text{Minimaliseer } \sum_m \text{cost}_m * \text{pmat}_m$$

Onder de voorwaarden:

$$lq_n \leq \sum_m qual_{mn} pmat_m \leq uq_n \text{ voor alle } n$$

$$pmat_m \leq urm_m \text{ voor alle } m$$

$$\sum_m pmat_m = 1.0$$

Waarbij de pmat variabelen de proportie van de betreffende grondstof voorstelt.

qual zijn de verschillende eigenschapskengetallen en lq en uq zijn onder- resp. bovengrens voor deze eigenschappen in het eindprodukt.

De bovengrens voor de proportie van een grondstof in het eindprodukt is urm.

De prijs van de grondstof is cost, en daarvoor wordt genomen het maximum van de voorraadprijs en de koopprijs. De grootte van de voorraad speelt dus geen rol.

Op verschillende plaatsen in het volgende wordt verwezen naar Fortran Conventies. Die worden beschreven in het volgende hoofdstuk.

In deze MGG specificaties onderscheiden we: hoofdregels, detailregels en vervolgregels.

Hoofdregels beginnen in positie 1.

Detail- en vervolgregels beginnen in positie 2 of later, behalve bij de ELEMENTS en de FUNCTIONS, waar ze de Fortran conventies volgen.

```

C
NOTATION
SUFFICES
C      M  MMAX  RAW MATERIAL          .
          50
C      N  NMAX  QUALITY
          20
VARIABLES
C      PMAT(M)  PROPORTION OF MATERIAL USED (0 < PMAT < 1.0)
                '*****MM'          NOT IF (URM(M).LE.0.0)
C      MATERIAL LIMITS
BOUND UP  B01          NOT IF (URM(M).GE.1.0)
C
EXTERNAL VALUES
C      STOCK COSTS OF MATERIALS
          SCOST(M)
C      COST OF BOUGHT MATERIALS
          BCOST(M)
C      RAW MATERIAL ANALYSIS
          QUAL(M,N)
C      LOWER QUALITY LIMITS
          LQ(N)          F5.0
C      UPPER QUALITY LIMITS
          UQ(N)
C      UPPER MATERIAL LIMITS (AS PROPORTION )
          URM(M)
C      MAXIMUM QUALITY VALUE
          XLARGE
C
PROBLEM
C
MINIMISE
*TCOST '*****'
          SUM(M) C01*PMAT(M)
SUBJECT TO
C      QUALITY LIMITS
*QULIM '***** NN'          NOT IF (LQ(N).LE.0.0.AND.UQ(N).GE.XLARGE)
          SUM(M) C02*PMAT(M) .GE. C03
RANGE  R01
FOR ALL N
C
          PRODUCTION
C
*PROD  '****'
          SUM(M) PMAT(M) .EQ. C04
C
ELEMENTS
C      DEFINE COEFFICIENTS
          B01 = URM(M)
          C01 = FCOST()
          C02 = QUAL(M,N)
          C03 = LQ(N)
          R01 = UQ(N) - LQ(N)
          C04 = 1.0
FUNCTIONS
          FUNCTION FCOST()
C
C      SELECT LARGER OF STOCK AND BOUGHT COSTS
C      (I.E. USE REPLACEMENT COST IF IT EXCEEDS THE STOCK COST)
C
          FCOST = SCOST(M)
          IF (BCOST(M) .GT. SCOST(M)) FCOST = BCOST(M)
          RETURN
          END
ENDATA

```


10.3. Uitleg van het voorbeeld:

De regels met een C in positie 1 zijn commentaar regels. Gebruik ze voor de duidelijkheid!

De NOTATION hoofdregel is verplicht en geeft het begin van de specificaties aan.

De volgorde uit het voorbeeld dient ook aangehouden te worden.

10.4. SUFFICES

De SUFFICES hoofdregel geeft aan dat daarna de detailregels volgen voor de indices van het probleem. De detailregels bevatten achtereenvolgens en gescheiden door een of meerdere spaties:

- de suffix (index) 1 positie groot (verplicht)
- de naam (volgens Fortran Conventies) van het geldende maximum voor deze suffix (verplicht)
- de naam (Fortran) van het absolute maximum (niet verplicht)
- het absolute maximum (geheel getal) (verplicht).

(Later bij de invoer van ruwe gegevens, wordt de waarde van het dan geldende maximum voor de betrokken run ingevuld).

10.5. De VARIABLES

In de detailregel worden de naam (max. 5 posities) en de suffices (max. 7, tussen haakjes, gescheiden door komma's) gegeven. Een suffix is niet verplicht.

Daarna volgt een aanduiding hoe deze variabele in het MPS formaat er uit gaat zien (tussen ' tekens). Deze MPS naam is maximaal 8 tekens, met één of meerdere * wordt aangegeven dat deze posities overgenomen worden uit de naam. Met het teken van de betrokken suffix (zodanig herhaald) wordt aangegeven dat hier de teller van de suffix wordt ingevuld (bij meer mogelijkheden dan in de posities passen, worden ook letters gebruikt).

Spaties en niet ingevulde posities in de MPS naam worden door MGG voorzien van een . op de overeenkomstige plaats in de MPS naam.

De suffices in de naam en de MPS-naam dienen overeen te stemmen in naam en volgorde. Let hier met name op!

Tenslotte is er nog de NOT IF conditie specificatie (niet verplicht) voor die gevallen waarin de variabele niet bestaat. Hierbij wordt gebruik gemaakt van Fortran conventies voor condities. Ook kan gebruik gemaakt worden van nog niet hiervoor gedefinieerde externe waarden (ruwe gegevens). Gebruik hier geen coëfficiënten (zie later).

Binnen de VARIABLES gegevens moeten ook de BOUNDS opgegeven worden bij de betrokken variables. Deze kaart regel BOUND in de eerste 5 posities en vanaf pos. 7 gescheiden door een of meer spaties: het type (UP, LO, FX, UI, SC, MI, FR of BV, zie MPS beschrijving), de coëfficiënt (niet bij type MI, FR of BV) en zo nodig een NOT IF conditie.

De coëfficiënt wordt i.h.a. pas in een later gedeelte van de MGG invoer gespecificeerd, hier wordt alleen de naam opgegeven. Deze naam moet uit precies 3 tekens bestaan, die een naam of een getal aangeven. De naam moet dan in de ELEMENTS sectie van de invoer nader gedefinieerd worden.

Soms is er meer dan één BOUND per variabele, mogelijke combinaties zijn:

LO en UP , MI en UP , LO en UI.

10.6. EXTERNAL VALUES

Hiermee worden de ruwe gegevens die later ingevoerd moeten worden bedoeld.

Eerst wordt de (Fortran type) naam gegeven met eventueel suffices tussen haakjes, daarachter volgt eventueel de format specificatie In of Fm.n

Default, bij niet specificatie, is I5 of F5.1, afhankelijk van de naam. Denk aan de type conventie voor Fortran namen.

Als dit type niet klopt met het format geeft MGG een waarschuwing en verandert het type. (Dit zal dus gebeuren bij LQ (N) in het voorbeeld).

10.7. De PROBLEM hoofdregel geeft aan dat de NOTATION over is.

Daarna moet MINIMIZE of MAXIMIZE opgegeven worden (verplicht).

Let op! MGG verandert MAXIMIZE situaties in MINIMIZE door het teken van de kostfunctie te veranderen. Daarom moet bij SCICONIC dan niet ook SETUP (MAXIMIZE) gegeven worden want dan gebeurt dit dubbel (omdat SCICONIC dat ook al doet).

Gebruik daarom bij MGG altijd MINIMIZE en geef zonodig MAXIMIZE aan bij SCICONIC.

In de volgende regel staat in positie 1 een *. Dit betekent dat hier de doelfunctie gegeven wordt. De naam van de objectfunctie is maximaal 5 posities lang.

Daarna volgt de MPS-naam, die in het voorbeeld gelijk is aan de doelfunctie naam.

In de volgende detailregel(s) wordt de specificatie van de doelfunctie gegeven.

Let op dat ook hier weer een coëfficiënt (3 tekens) gebruikt moet worden (COI) die pas later gedefinieerd wordt.

I.h.a. kunnen meerdere termen (al of niet met SUM functies) opgeteld of afgetrokken worden om de gehele doelfunctie te krijgen. Waar nodig kunnen vervolgregels (pos. 1 leeg) gebruikt worden.

De SUBJECT TO regel geeft aan dat hierna de normale constraints volgen, die ook in positie 1 weer een * hebben. Na de * volgt dan de constraint naam (max. 5 posities) met de indices en daarna de MPS naam (let op dat de indices kloppen!).

Bovendien kan hier een NOT IF conditie volgen.

De detailregels voor de constraints bevatten de termen bestaande uit +, - of SUM, een coëfficiënt * een variabele naam.

Daarna wordt de constraint soort aangegeven met .GE of .LE of .EQ die resp. type G, L of E in MPS format worden.

Tenslotte de RHS waarde via een coëfficiënt.

In ons voorbeeld, waar zowel een boven- als een ondergrens geldt voor de constraints, wordt een Range gegeven met alweer een coëfficiënt. Hier kan ook nog een NOT IF conditie bij staan.

Tenslotte de FOR ALL regel voor overgebleven suffices uit de constraint definitie.

10.8. ELEMENTS

Hierna volgen de ELEMENTS regels waarin nu pas alle gebruikte coëfficiënten moeten worden gedefinieerd. Dit gaat via Fortran type statements (beginnen in positie 7).

Ook Functions kunnen gebruikt worden (denk eraan, altijd lege haakjes te gebruiken bij een function).

De FUNCTIONS geven de mogelijkheid nog complexere Fortran type instructies te gebruiken. Eventuele parameters hoeven niet tussen de haakjes meegegeven te worden. Het MGG systeem zorgt daar al voor (m.b.v. COMMON structuren). Functions altijd afsluiten met RETURN en END. (Denk er aan te beginnen in positie 7).

De MGG invoer altijd afsluiten met de ENDDATA regel.

10.9. Verboden namen

In verband met het interne gebruik daarvan door MGG zelf, zijn de volgende namen niet toegestaan voor uw gebruik:

BIG	INUM	LMGINP	MKINPT
BLKCOM	IOBJST	LMGINT	MKNAME
CLINIT	KBLANK	LMGNAM	MKROW
CAFIND	KCARD	LMGNUM	MKTITLE
CAGET	KCNCLR	LMGVAL	MKVERS
CAPUT	KCNCLW	LRWCOL	MSEQ
CEND	KCOMM	LRWNXT	MXERR
CIFIND	KDATA	LRWOBJ	MXLINE
CIGET	KIDENT	LRWROW	NCARD
CINBLK	KMATR	LSBIT	NCOMP
CINIT	KPRINT	LSBYT	NERROR
CIPUT	KREAD	LWCOMP	NEXVEC
CPRIUP	KTERR	MABS	NLINE
CRFIND	LACOMP	MBLOCK	NNUM
CRGET	LAGBIT	MGBND	NPAGE
CRPUT	LASBIT	MGCOL	NVERNO
DATE	LCBYTS	MGCOMM	OBJVAL
F8SLRD	LCFILL	MGGREP	POSIT
GETCLL	LDEBUG	MGINFO	RNUM
GETFIL	LDREAD	MGOPT	RVAL
IADDR	LERROR	MGMATR	RWCOMM
IANUM	LGBIT	MGMAXS	RWREAD
IFERR	LGBYT	MGRHS	SMALL
IFMORE	LIHEAD	MGRNG	STCOL
IFSAME	LMGCMF	MGRW	STROW
IHEAD	LMGCRD	MGSET	TIME
ININIT	LMGFLD	MGUSER	
INREAD	LMGID	MKCOL	

Bovendien is het onverstandig de volgende, door Fortran gebruikte, namen te gebruiken:

ABS	ENTRY	MAIN	READ
CALL	FUNCTION	MAX	RETURN
COMMON	GOTO	MIN	SAVE
DATA	IF	MOD	STOP
DO	INTEGER	PAUSE	WRITE
END	LOGICAL	READ	

Tenslotte is het de verantwoordelijkheid van de gebruiker te zorgen dat verschillende namen bij invulling van de sufficer ook uniek blijven.

10.10 MGG uitvoeren

Dit gaat in een aantal stappen:

1. Maak een bestand met de MGG invoer (specificaties) volgens de gegeven beschrijving.

2. Type MGG file/OUTPUT = dest.
Vul hierbij voor file de naam in van het bestand met de specificaties invoer. (default type is .DAT). Vul voor de dest de naam van een file in waarop de uitvoer van de MGG run komt (listing, waarschuwing, fout boodschappen etc.). Default type is .LIS . Op deze file komt ook de beschrijving van de invoer voor de volgende fase. Zorg dat u deze file in geprinte vorm kunt krijgen. U kunt ook de uitvoer direct op uw terminal krijgen door voor dest TT: in te vullen. Default (als /OUTPUT=dest weggelaten wordt) is de file MGGOP.LIS.
3. Type MGCL om de gegenereerde Fortran programma's te vertalen, enz.
4. Maak de ruwe data invoerfile volgens de specificaties uit de uitvoerfile van MGG.
5. Type MG file/OUTPUT=dest.
Vul voor file in de naam van de datafile uit 4 (default type is .DAT). Als geen naam gegeven wordt dan wordt MGDATA.DAT gebruikt. Voor dest kunt u weer een file invullen die de uitvoer listing bevat, of TT: voor de terminal uitvoer. Default is MGOP.LIS.
6. Als alles goed gegaan is, wordt de matrix gezet op de file MATRIX.DAT.
Deze file kan nu als invoer voor SCICONIC (b.v. m.b.v. LPSOLVE) gebruikt worden.
7. Er worden door MGG nogal wat files aangemaakt die na afloop verwijderd kunnen worden. Hiertoe kunt u het commando CLEAN intypen. Uiteraard kunt u m.b.v. DELETE ook zelf een voor een files verwijderen.

MGG S80085.DAT
MGG EXECUTED SUCCESSFULLY
MGG
TYPE MGGOP.LIS

MGCL hoeft niet voor de TYPE MGGOP.LIS
te komen, mag ook daarna.

SCICON MGG VERSION 01.06VM

HARD INPUT

```
1. C      VOORBEELD UIT DE SYLLABUS LP BLDE 30
2. NOTATION
3. SUFFICES
4.   P PMAX 10
5.   M MMAX 10
6. VARIABLES
7.   PRODX(P) '*****P'
8. EXTERNAL VALUES
9.   CAPAC(M)
10.  BNODIG(P,M)
11.  WINST(P)
12. PROBLEM
13. MINIMISE
14. *WINST '*****'
15.  SUM(P) C01 * PRODX(P)
16. SUBJECT TO
17. *MACH '**** MM'
18.  SUM(P) C02 * PRODX(P) .LE. C03
19. FOR ALL M
20. ELEMENTS
21.      C01 = WINST(P)
22.      C02 = BNODIG(P,M)
23.      C03 = CAPAC(M)
24. ENDDATA
```

Dit is hetzelfde voorbeeld uit de syllabus L.P.
dat ook als voorbeeld gebruikt is voor het
MPS formaat (in hoofdstuk 3.1) en bij het
gebruik van LPSOLVE (in hoofdstuk 8)

U ziet dat relatief veel moeite nodig is voor zo'n klein
probleem. Het is nu echter zeer gemakkelijk geworden
om de grootte van het probleem sterk uit te breiden.

```
*****  
***MG DATA INPUT***  
*****
```

26-AUG-85 12:27:48

NOTES ON DATA INPUT

1. ALL SEQUENCE NUMBERS AND INDEX NUMBERS SHOULD BE RIGHT JUSTIFIED.
2. IMPLIED INDEX VARIES ALONG THE CARD. EXTERNAL INDICES VARY OVER SEQUENCE NUMBERED SETS OF CARDS.
3. DATA FIELDS SHOULD END ON POSITIONS MARKED '*'.
*
4. A STAR IN A COLUMN POSITON AFTER THE LAST REQUIRED POSITION INDICATES THAT THE COLUMN IS TO BE IGNORED.
5. FOR ALL DEFINITIONS SEE USERS GUIDE.
6. ORDER OF CARD INPUT SHOULD BE AS FOLLOWS:

OPTIONAL CARDS:

COLUMNS
1234567

OPTION (FREE FORMAT OPTIONS CARD)
* (COMMENT CARD)
TITLE (FREE FORMAT PROBLEM TITLE CARD)
VERSION (FREE FORMAT VERSION NUMBER AND NAME CARD)
UNITS (FREE FORMAT CARD TO ALTER TOLERANCES)
MAXIMA OR
MAXIMUMS (FREE FORMAT CARD TO ALTER THE CURRENT MAXIMA OF SUFFICES)

COMPULSORY CARDS:
(SEE FOLLOWING PAGES)

NAME
EXTERNAL VALUES

CARD TYPE
HEADER CARD

FORMATTED (64+2) 15 13F5.0)
1111111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

DATA GENERATOR EXPECTS 1 CARD WITH THE NAME EXTERNAL VALUES

NAME
CAPAC

IMPLIED INDEX
M

EXTERNAL INDEX
NONE

FORMATTED (64+2) 15 13F5.0)
1111111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

DATA GENERATOR EXPECTS UP TO 1 CARDS WITH THE NAME CAPAC CONTAINING A
MAXIMUM OF 10 DATA FIELDS PER CARD

NAME
ENODIG

IMPLIED INDEX
M

EXTERNAL INDEX
P

FORMATTED (64+2) 15 12F5.0)
1111111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

DATA GENERATOR EXPECTS UP TO 10 CARDS WITH THE NAME ENODIG CONTAINING A
MAXIMUM OF 10 DATA FIELDS PER CARD

NAME
ENINT

IMPLIED INDEX
P

EXTERNAL INDEX
NONE

FORMATTED (64+2) 15 13F5.0)
1111111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

DATA GENERATOR EXPECTS UP TO 1 CARDS WITH THE NAME ENINT CONTAINING A
MAXIMUM OF 10 DATA FIELDS PER CARD

Dit is de gebruikte ruwe data invoer.

BUICON MGG VERSION 01.06VM

CARD INPUT

1.	TITLE	DATA UNDER NAME SYLLABUS LP BLKZ 30				
2.	UNITED	SYLLABUS				
3.	MAXIMA	P MAX=2 M MAX=4				
4.	EXTERNAL	VALUES				
5.	CAPAC	1	20	10	30	24
6.	ENDDIG	1	1	1	1	3
7.	SNDDIG	2	1	4	2	3
8.	WINST	1	1	1		

Dit is de geproduceerde matrix:

BUICON MGG VERSION 01.06VM

CARD INPUT

PROBLEM STATISTICS:

5 ROWS

2 COLUMNS

10 NON-ZERO ELEMENTS

100.00 PERCENT DENSITY

NO RAN OK

\$ TYPE MATRIX.DAT

NAME SYLLABUS

ROWS

N WINST...

L MACH.01.

L MACH.02.

L MACH.03.

L MACH.04.

COLUMNS

PRODX1.. WINST... 1.000000 MACH.01. 1.000000

PRODX1.. MACH.02. 1.000000 MACH.03. 2.000000

PRODX1.. MACH.04. 3.000000

PRODX2.. WINST... 1.000000 MACH.01. 4.000000

PRODX2.. MACH.02. 2.000000 MACH.03. 3.000000

PRODX2.. MACH.04. 2.000000

RHS

RHSSET01 MACH.01. 20.000000 MACH.02. 12.000000

RHSSET01 MACH.03. 30.000000 MACH.04. 24.000000

ENDATA

\$

MGG S30ERR
G EXECUTED SUCCESSFULLY
TYPE MGGOP.LIS

maar er zijn wel WARNINGS!

SCICON MGG VERSION 01.06VM

CARD INPUT

PAGE 1

```
1. C      VOORBEELD UIT DE SYLLABUS LP BLDZ 30
2. NOTATION
3. SUFFICES
4.   P PMAX 10
5.   M MMAX 10
6. VARIABLES
7.   PRODX(P) '*****P'
8. EXTERNAL VALUES
9.   CAPAC(M)
10.  BNODIG(P,M)
11.  WINST(P)
12. PROBLEM
13. MINIMISE
14. *WINST '*****'
15.  SUM(P,M) C01 * PRODX(P)
```

WARNING 617 'SUM' SUFFIX DOES NOT APPEAR IN VARIABLE OR LP NAME SUFFICES
hiermee wordt verwezen naar de M in de SUM

```
16. SUBJECT TO
17. *MACH '**** MM'
18.  SUM(P,M) C02 * PRODX(P) .LE. C03
```

WARNING 617 'SUM' SUFFIX DOES NOT APPEAR IN VARIABLE OR LP NAME SUFFICES

WARNING 627 SUFFIX IN SUM IS ALSO IN LP NAME

```
19. FOR ALL M
20. ELEMENTS
21.   C01 = WINST(P)
22.   C02 = BNODIG(P,M)
23.   C03 = CAPAC(M)
24. ENDDATA
```

*vergelijk deze run met fouten met de
hiervoor gegeven goede run!*

11. Fortran conventies

11.1. Namen

Een aantal eigenschappen van MGG zijn terug te voeren tot Fortran regels en conventies. De belangrijkste volgen hier:

- Namen mogen niet langer dan 6 tekens zijn en moeten met een letter beginnen.
- ⇒ Namen beginnend met een I, J, K, L, M of N zijn type integer (geheel getal) tenzij anders aangegeven.
- ⇒ Alle andere namen zijn type real tenzij anders aangegeven.

11.2. Formats

Beschrijving van de precieze invoerformaten en posities gaat via het FORMAT. Dit bevat tussen komma's de aanduiding voor type en grootte van het veld:

Zo is nX (n is elk geheel getal), een aanduiding dat n posities overgeslagen moeten worden.

In is een geheel getal in de volgende n posities.

An is een tekst (willekeurige tekens) in de volgende n posities.

Fn.m is een reëel getal van n posities met m daarvan achter de punt (geen komma!).

Zowel voor de I als de A en de F mag nog een geheel getal staan, dat aangeeft hoeveel keer deze specificatie herhaald moet worden.

11.3. Conditie

Een conditie is een vergelijking van twee vergelijkbare zaken, b.v. waarden, tussen haakjes.

(NAAM 1 .COND. NAAM2)

NAAM 1 en NAAM 2 kunnen een Fortran NAAM zijn, of een getal.

De mogelijke conditie types zijn:

- .EQ. is gelijk aan
- .GT. is groter dan
- .GE. is groter of gelijk
- .LT. is kleiner dan
- .LE. is kleiner of gelijk

.NE. is ongelijk aan.

Meervoudige condities kunnen met .AND. en .OR. gebouwd worden. Gebruik extra haakjes voor de duidelijkheid!

11.4. Statements

Fortran statements beginnen in positie 7 behalve vervolg kaarten die een teken (niet 0) in positie 6 hebben.

Als er een label (verwijzingsnummer) gebruikt wordt staat dat voor de regel in positie 2 t/m 5.

12. Fouten en foutmeldingen

De mogelijkheden tot het maken van fouten bij het gebruik van de Mathematische Programmering software zijn legio.

Sommige fouten zullen per definitie niet door het systeem ontdekt kunnen worden (b.v. een niet bedoelde naam, een niet bedoelde constraint, etc.).

Het systeem kan natuurlijk wel andere fouten melden en waarschuwingen geven, de duidelijkheid van zulke meldingen is echter afhankelijk van de situatie. Het is een bekend verschijnsel dat een fout kan leiden tot meerdere foutmeldingen, omdat dan het systeem de draad van uw bedoeling kwijt is.

Verschillende fasen in de verwerking kunnen foutmeldingen genereren. Om de juiste correctie te kunnen doen is het van belang eerst te bepalen uit welke stap de foutmelding komt.

1. THENET Te herkennen aan de # prompt. De verschillende foutmeldingen zijn te vinden in rapport AG-64.
2. VAX-VMS Hier kunnen zeer veel foutmeldingen uit komen. Ze zijn herkenbaar aan een vreemde naamcode vóór de Engelstalige boodschap.
Meestal zal de fout te maken hebben met files (die niet bestaan of iets dergelijks).
Zie ook het voorbeeld.
3. Nederlandstalige foutboodschappen komen uit de speciaal gemaakte procedures en programma's. Deze zouden duidelijk moeten zijn.
4. SCICONIC Hier kunnen foutmeldingen (zie voorbeeld), maar ook opmerkingen (problem infeasable) vandaan komen.
5. MGG Hier bestaat de grootste mogelijkheid tot het maken van fouten. Een aantal tips in verband met de meest voorkomende fouten volgt hier.

- Volgorde van suffices: X(I,J) '*JJII' is fout en moet zijn:

X(I,J) '*IIJJ'

- Minstens één external value moet aanwezig zijn.

- Gebruik niet de naam voor het huidige maximum als external value.

- Suffices moeten kloppen:

b.v. in een object: SUM(M,P) CO3*PROD(P)

hier is een suffix (M) te veel, als er in CO3 geen M voorkomt

- Verwar het cijfer nul niet met de letter o.

- In de foutboodschappen van MGG wordt over de LP name gesproken waar de MPS naam bedoeld wordt.

- Zet niet MAXIMIZE in MGG en in SCICONIC

Sommige MGG fouten worden pas tijdens de compilatie fase ontdekt door de Fortran comiler, of in de LINK of executie fase door het VMS systeem.

Tenslotte nog dit: Het is onmogelijk alle fouten die op kunnen treden te voorzien.

Ervaring is de beste leermeester.

U wordt daarom verzocht zeldzame of bijzonder fraaie fouten of foutmeldingen te verzamelen voor de begeleiders en om deze handleiding daarmee uit te breiden.

Wat is het invoerbestand met de MATRIX in MPS formaat?

(Default is MATRIX.DAT)

Als u deze file nu eerst wilt maken type dan : INPUT

INPUT

Hoe moet de te maken file heten ? (default MATRIX.DAT)

MPSDATA

Default type is .DAT dus MPSDATA.DAT

===== M P S F O R M A A T I N V O E R G E N E R A T O R T H E =====

Geef een naam aan het LP model. (max. 8 tekens) (default is LPMODEL)

Niets ingetoetst, daarom de default naam gebruikt

LPMODEL

Geef de naam voor de DOEL-FUNCTIE (max 8 tekens) (default is OBJECT)

WINST

Geef de naam van een CONSTRAINT rij (max. 8 tekens)

als alle namen seseven zijn dan alleen RETURN toets

MACHINE1

Wat is het TYPE van de constraint (N, G, E of L)

L

Geef de naam van een CONSTRAINT rij (max. 8 tekens)

als alle namen seseven zijn dan alleen RETURN toets

MAXINE2

Deze typpfout kan niet ontdekt worden

Wat is het TYPE van de constraint (N, G, E of L)

K

Deze typpfout wordt wel ontdekt

Kies N(ND), G(GREATER THAN), E(EQUAL) of L(LESS THAN)

Wat is het TYPE van de constraint (N, G, E of L)

L

Geef de naam van een CONSTRAINT rij (max. 8 tekens)

als alle namen seseven zijn dan alleen RETURN toets

MACHINE3

Wat is het TYPE van de constraint (N, G, E of L)

L

Geef de naam van een CONSTRAINT rij (max. 8 tekens)

als alle namen seseven zijn dan alleen RETURN toets

MACHINE3

Deze constraint-naam is al seseven !

Geef de naam van een CONSTRAINT rij (max. 8 tekens)

als alle namen seseven zijn dan alleen RETURN toets

?

Hier door krijg ik de lijst van tot nu toe gebruikte

Bestaande constraint-namen en type :

constraint namen

WINST	N
MACHINE1	L
MAXINE2	L
MACHINE3	L

Geef de naam van een CONSTRAINT rij (max. 8 tekens)

als alle namen seseven zijn dan alleen RETURN toets

MACHINE4

Wat is het TYPE van de constraint (N, G, E of L)

L

Geef de naam van een CONSTRAINT rij (max. 8 tekens)

als alle namen seseven zijn dan alleen RETURN toets

Gereed met de constraints, dus niets ingetoetst, alleen RETURN

Geef de naam van een VARIABELE(kolom)(max 8 tekens)

als alle namen seseven zijn dan alleen RETURN toets

XXXXX

Geef de naam van een VARIABELE(kolom)(max 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets
PRODX2

Geef de naam van een VARIABELE(kolom)(max 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets

? Nu krijg ik te zien welke variabelen er zijn

Bestaande variabele-namen :
PRODX1
PRODX2

Geef de naam van een VARIABELE(kolom)(max 8 tekens)
als alle namen seseven zijn dan alleen RETURN toets
gereed met de variabelen

Geef de RIGHT HAND SIDE naam, of alleen RETURN (default is RHS)
UREN

Geef de RIGHT HAND SIDE naam, of alleen RETURN (default is RHS)
geen verdere right hand sides

Geef een RANGE naam, of alleen RETURN toets
geen ranges

Geef de naam van een BOUND, of alleen RETURN toets
geen bounds

Wilt u de matrix elementen ZELF OPGEVEN (JA)
of ze een voor een laten OPVRAGEN (NEE)

NEE Als de matrix groot en leeg is, is zelf opgeven te prefereren

Kies invoer volgorde, per CONSTRAINT of VAR (C/V)

C

Constraint : WINST

Geef de waarde bij variabele : PRODX1

1 getallen mogen als gehele waarden gegeven worden

Geef de waarde bij variabele : PRODX2

1.0 mag ook als reële waarde met decimale punt

Constraint : MACHINE1

Geef de waarde bij variabele : PRODX1

1 Geef de waarde bij variabele : PRODX2

4 Geef de waarde in rhs : UREN

20 Geef de waarde in rhs : UREN

Constraint : MAXINE2

Geef de waarde bij variabele : PRODX1

1 Geef de waarde bij variabele : PRODX2

2 Geef de waarde in rhs : UREN

12 Geef de waarde in rhs : UREN

Constraint : MACHINE3

Geef de waarde bij variabele : PRODX1

2 Geef de waarde bij variabele : PRODX2

3 Geef de waarde in rhs : UREN

30 Geef de waarde in rhs : UREN

Constraint : MACHINE4

Geef de waarde bij variabele : PRODX1

3 Geef de waarde bij variabele : PRODX2

2 Geef de waarde in rhs : UREN

24 Geef de waarde in rhs : UREN

2 KOLOMMEN

10 NIET-NUL ELEMENTEN

100.00 PROCENT DICHTHEID

FORTRAN STOP

Wilt u de zojuist semaeke matrix bekijken ? (J of N)

JA

NAME	LPMODEL		
ROWS			
N	WINST		
L	MACHINE1		
L	MAXINE2		
L	MACHINE3		
L	MACHINE4		
COLUMNS			
PRODX1	WINST	1.00000	
PRODX1	MACHINE1	1.00000	
PRODX1	MAXINE2	1.00000	
PRODX1	MACHINE3	2.00000	
PRODX1	MACHINE4	3.00000	
PRODX2	WINST	1.00000	
PRODX2	MACHINE1	4.00000	
PRODX2	MAXINE2	2.00000	
PRODX2	MACHINE3	3.00000	
PRODX2	MACHINE4	2.00000	
RHS			
UREN	WINST	1.00000	
UREN	MACHINE1	20.0000	
UREN	MAXINE2	12.0000	
UREN	MACHINE3	30.0000	
UREN	MACHINE4	24.0000	

ENDATA

Wilt u nu met SCICONIC de LP oplossins vinden? (J of N)

JA

Wat is de SCICONIC uitvoer file ?(Default RESULT.DAT)

UITVR

Specificeer MAXIMISE of MINIMISE svp (MAX of MIN)

MAX

Wilt u een INTERACTIEVE of een BATCH run ? (I of B)

het default antwoord is I hierop

Is dit een Continu of een Integer run ? (C of I)

het default antwoord is C hierop

SCICONIC/VM VERSION VM/V1.30
COPYRIGHT SCICON LTD. 1983

AUTHORISED FOR USE AT:
UNIVERSITY OF TECHNOLOGY EINDHOVEN

1I>

1I>

ON-ERROR GOTO 9

2I>

CONVERT

PROBLEM LPMODEL - NEW VERSION

RHS VECTOR - UREN

PROBLEM HAS 5 ROWS, 2 COLUMNS AND 10 NON-ZERO ELEMENTS

Hier begint de interactieve uitvoer van SCICONIC.

Echter, niet echt interactief, want door de eerdere vragen zijn al SCICONIC commando's al gegeneerd. U ziet wel wat er gebeurt.

Als er fouten gevonden worden ga dan naar label 9

```

3I>
SETUP(MAXIMISE)
PROBLEM LPMODEL ON FILE PROBLEM
CREATED ON 30-AUG-1985 09:36:14.23
PROBLEM HAS 5 ROWS, 2 COLUMNS AND 10 NON-ZERO ELEMENTS
RHS - UREN
OBJECTIVE - WINST
INCORE MATRIX HAS 5 ROWS AND 2 COLUMNS
SETUP TOOK 0.86 SECONDS

```

```

4I>
PRIMAL

```

NITS	OBJECT	INFEAS	SECS
0	1.000000	0.000000(0)	1.02
2	-8.000000	0.000000(0)	1.09

SOLUTION IS OPTIMAL

```

5I>
PRINTSOLN

```

```

PROBLEM LPMODEL - SOLUTION NUMBER 1 - OPTIMAL
CREATED ON 30-AUG-1985 09:36:17.47 , AFTER 2 ITERATIONS
PRINTED ON 30-AUG-1985 09:36:18.04

```

...NAME...	..ACTIVITY..	DEFINED AS
FUNCTIONAL RESTRAINTS	8.000000	WINST UREN

```

?:
?:

```

..ROW...	ATACTIVITY....
N WINST	BS	-9.000000
L MACHINE1	BS	18.000000
L MAXINE2	UL	12.000000
L MACHINE3	BS	21.000000
L MACHINE4	UL	24.000000

```

*** END OF ROWS ***
?:

```

.COLUMN.	ATACTIVITY....
PRODX1	BS	6.000000
PRODX2	BS	3.000000

```

*** END OF COLUMNS ***
?:

```

```

6I>
PUNCHSOLN

```

```

7I>
9 CONTINUE

```

```

8I>
STOP

```

*Nu in de verwerking
beëindigd*

LPMODEL

PROBLEM LPMODEL - SOLUTION NUMBER 1 - OPTIMAL

CREATED ON 22-AUG-1985 15:50:56.98 , AFTER 2 ITERATIONS

PRINTED ON 22-AUG-1985 15:51:01.15

..NAME...	..ACTIVITY..	DEFINED AS
FUNCTIONAL	8.000000	WINST
RESTRAINTS		UREN

LPMODEL

BER	..ROW...	ATACTIVITY....	.SLACK	ACTIVITY.	..LOWER BOUND...	..UPPER BOUND...	.DUAL ACTIVITY..
1	WINST	BS	-9.000000	8.000000		NONE	NONE	1.000000
2	MACHINE1	BS	18.000000	2.000000		NONE	20.000000	.
3	MACHINE2	UL	12.000000	.		NONE	12.000000	0.250000
4	MACHINE3	BS	21.000000	9.000000		NONE	30.000000	.
5	MACHINE4	UL	24.000000	.		NONE	24.000000	0.250000

LPMODEL

BER	.COLUMN.	ATACTIVITY....	...INPUT COST...	..LOWER BOUND...	..UPPER BOUND...	..REDUCED COST..
6	PRODX1	BS	6.000000	-1.000000	.	NONE	.
7	PRODX2	BS	3.000000	-1.000000	.	NONE	.

```
T-W-NOTSET, error modifying HSC000#DUA1:
I-E-IVDEVTYPE, invalid device type - specify a mailbox device
MATH.PROG.
ASSIGN USER2:[CLP]HELPPFILE.DAT HELPLINP
P := $USER2:[CLP]SCICONIC.EXE
PSOLVE := @USER2:[BDOR2]LPSOLVE
USER2:[CLP.MGG]MGGDEF
```

SYMBOL DEFINITIONS FOR MGG

```
ASSIGN user2:[1p.mss] MGG$DIR
MGG:=="@MGG$DIR:MGG/OUTPUT=SYS$OUTPUT SSS"
MGCCL:=="@MGG$DIR:MGCCL.COM"
MGCWL:=="@MGG$DIR:RWL.COM"
MGS:=="@MGG$DIR:MG/OUTPUT=SYS$OUTPUT SSS"
MGR:=="@MGG$DIR:RW/OUTPUT=SYS$OUTPUT SSS"
```

```
EXIT
EXIT
DEFAULT [BDOR2]
-I MATRIX -O RESULT -P PROBLEM
```

```
SCICONIC/VM VERSION VM/V1.30
COPYRIGHT SCICON LTD, 1983
```

AUTHORISED FOR USE AT:
UNIVERSITY OF TECHNOLOGY EINDHOVEN

*Hier begint de
SCICONIC uitvoer*

```
>
INVERT
INVERT
INVERT
DBLEM LPMODEL - NEW VERSION
S VECTOR - UREN
DBLEM HAS 5 ROWS, 2 COLUMNS AND 10 NON-ZERO ELEMENTS
INVERT TOOK 0.49 SECONDS
```

```
>
TUP(MAXIMISE)
TUP(MAXIMISE)
DBLEM LPMODEL ON FILE PROBLEM
CREATED ON 26-AUG-1985 11:46:42.31
DBLEM HAS 5 ROWS, 2 COLUMNS AND 10 NON-ZERO ELEMENTS
S - UREN
OBJECTIVE - WINST
CORE MATRIX HAS 5 ROWS AND 2 COLUMNS
TUP TOOK 0.76 SECONDS
```

```
>
ERROR GOTO 9
ERROR GOTO 9
```

```
>
IMAL
IMAL
```

NITS	OBJECT	INFEAS	SECS
0	1.000000	0.000000(0)	0.88
2	-8.000000	0.000000(0)	0.95

SOLUTION IS OPTIMAL

```
>
INTSOLN
INTSOLN
```

PROBLEM LPMODEL - SOLUTION NUMBER 1 - OPTIMAL

CREATED ON 26-AUG-1985 11:46:44.53 , AFTER 2 ITERATIONS

PRINTED ON 26-AUG-1985 11:46:44.77

...NAME...	..ACTIVITY..	DEFINED AS
FUNCTIONAL	8.000000	WINST
RESTRAINTS		UREN

LPMODEL

PAGE 2

NUMBER	..ROW...	ATACTIVITY....	.SLACK	ACTIVITY.	..LOWER BOUND...	..UPPER BOUND...	.DUAL ACTIVITY..
1	WINST	BS	-9.000000		8.000000	NONE	NONE	1.000000
2	MACHINE1	BS	18.000000		2.000000	NONE	20.000000	.
3	MACHINE2	UL	12.000000		.	NONE	12.000000	0.250000
4	MACHINE3	BS	21.000000		9.000000	NONE	30.000000	.
5	MACHINE4	UL	24.000000		.	NONE	24.000000	0.250000

LPMODEL

PAGE 3

NUMBER	.COLUMN.	ATACTIVITY....	...INPUT COST...	..LOWER BOUND...	..UPPER BOUND...	..REDUCED COST..
6	PRODX1	BS	6.000000	-1.000000	.	NONE	.
7	PRODX2	BS	3.000000	-1.000000	.	NONE	.

CHSOLN
CHSOLN

CONTINUE
CONTINUE

CRAN STOP
ROR2

Job terminated at 26-AUG-1985 11:46:51.14

Accounting information:
 Offered I/O count: 74 Peak working set size: 1244
 Direct I/O count: 106 Peak page file size: 16742
 Page faults: 1218 Mounted volumes: 0
 Elapsed CPU time: 0 00:00:10.26 Elapsed time: 0 00:00:18.61

Einde van de Batch job

GG EXECUTED SUCCESSFULLY

MGL
TYPE MGGOP.LISMGL hoeft niet voor de TYPE MGGOP.LIS
te komen, mag ook daarna.

SCICON MGG VERSION 01.06VM

CARD INPUT

```

1.  C      VOORBEELD UIT DE SYLLABUS LP BLDZ 30
2.  NOTATION
3.  SUFFICES
4.    P PMAX 10
5.    M MMAX 10
6.  VARIABLES
7.    PRODX(P) '*****P'
8.  EXTERNAL VALUES
9.    CAPAC(M)
10.   BNODIG(P,M)
11.   WINST(P)
12.  PROBLEM
13.  MINIMISE
14. *WINST '*****'
15.   SUM(P) C01 * PRODX(P)
16.  SUBJECT TO
17. *MACH '**** MM'
18.   SUM(P) C02 * PRODX(P) .LE. C03
19.  FOR ALL M
20.  ELEMENTS
21.      C01 = WINST(P)
22.      C02 = BNODIG(P,M)
23.      C03 = CAPAC(M)
24.  ENDATA

```

Dit is hetzelfde voorbeeld uit de syllabus L.P. dat ook als voorbeeld gebruikt is voor het MPS formaat (in hoofdstuk 3.1) en bij het gebruik van LPSOLVE (in hoofdstuk 8)

U ziet dat relatief veel moeite nodig is voor zo'n klein probleem. Het is nu echter zeer gemakkelijk geworden om de grootte van het probleem sterk uit te breiden.

MG DATA INPUT

26-AUG-85 12:27:48

NOTES ON DATA INPUT

1. ALL SEQUENCE NUMBERS AND INDEX NUMBERS SHOULD BE RIGHT JUSTIFIED.
2. IMPLIED INDEX VARIES ALONG THE CARD. EXTERNAL INDICES VARY OVER SEQUENCE NUMBERED SETS OF CARDS.
3. DATA FIELDS SHOULD END ON POSITIONS MARKED '*'.
4. A STAR IN A COLUMN POSITON AFTER THE LAST REQUIRED POSITION INDICATES THAT THE COLUMN IS TO BE IGNORED.
5. FOR ALL DEFINITIONS SEE USERS GUIDE.
6. ORDER OF CARD INPUT SHOULD BE AS FOLLOWS:

OPTIONAL CARDS:

COLUMNS
1234567

OPTION (FREE FORMAT OPTIONS CARD)
* (COMMENT CARD)
TITLE (FREE FORMAT PROBLEM TITLE CARD)
VERSION (FREE FORMAT VERSION NUMBER AND NAME CARD)
UNITS (FREE FORMAT CARD TO ALTER TOLERANCES)
MAXIMA OR
MAXIMUMS (FREE FORMAT CARD TO ALTER THE CURRENT MAXIMA OF SUFFICES)

COMPULSORY CARDS:
(SEE FOLLOWING PAGES)

FORMAT(3A4, A3)

11111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
EXTERNAL VALUES*****

MATRIX GENERATOR EXPECTS 1 CARD WITH THE NAME EXTERNAL VALUES

NAME	IMPLIED INDEX	EXTERNAL INDEX
CAPAC	M	NONE

FORMAT: (A4,A2, I5, 13F5.0)

11111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
CAPAC -SEQ- * * * * * * * * * * *****

MATRIX GENERATOR EXPECTS UP TO 1 CARDS WITH THE NAME CAPAC CONTAINING A
MAXIMUM OF 10 DATA FIELDS PER CARD

NAME	IMPLIED INDEX	EXTERNAL INDEX
BNDIG	M	P

FORMAT: (A4,A2, I5, 12F5.0)

11111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
BNDIGPPPPP-SEQ- * * * * * * * * * * *****

MATRIX GENERATOR EXPECTS UP TO 10 CARDS WITH THE NAME BNDIG CONTAINING A
MAXIMUM OF 10 DATA FIELDS PER CARD

NAME	IMPLIED INDEX	EXTERNAL INDEX
WINST	P	NONE

FORMAT: (A4,A2, I5, 13F5.0)

11111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890
WINST -SEQ- * * * * * * * * * * *****

MATRIX GENERATOR EXPECTS UP TO 1 CARDS WITH THE NAME WINST CONTAINING A
MAXIMUM OF 10 DATA FIELDS PER CARD