III

INSTITUUT VOOR KERNPHYSISCH ONDERZOEK
AMSTERDAM

1973

SO-1

BAKALG, een bedradingsprogramma met optimalisatie van draadlengte

Jan Visschers
Pier ten Kate

INHOUD

1. Inleiding

Door de digitale groep wordt al enkele jaren gebruik gemaakt van het EL-X8 programma BAKSORT (geschreven door Hans Suys en Bob Wielinga) om bedradingslijsten voor komputer-interfaces te sorteren. Wegens het beschikbaar komen van steeds snellere komponenten, en ook omdat de interfaces groter en ingewikkelder worden, was een verbeterde sorteermetode noodzakelijk. Omdat dit een vrij ingrijpende wijziging van het bestaande programma betrof, is besloten het gehele programma om te zetten naar Algol-60. Dit heeft een aantal voordelen:

1. Er kon gebruik gemaakt worden van bestaande sorteer- en optimalisatie-procedures, waardoor de omzetting slechts twee manweken kostte.
2. BAKALG kan nu op bijna elke grote rekeninstallatie gedraaid worden, zoals op het Math. Centrum, op de SARA (via de terminal die binnenkort op het IKO komt) en op een eventuele opvolger van de X8.

Als nadeel staat hier tegenover een grotere rekentijd. Aangezien de frekwentie van gebruik van BAKSORT niet erg hoog is, weegt dit nadeel niet op tegen de genoemde voordelen.

2. Beschrijving van het probleem

Digitale elektronika wordt op het IKO vervaardigd in een standaard bouwsiesteem dat opgebouwd wordt uit konnektorblokken, waarin aan de voorkant gedrukte bedradingskaartjes met digitale komponenten worden gestoken. Aan de achterkant zijn de blokken voorzien van een groot aantal pennen, die met behulp van een "wrap" pistool paarsgewijs aan elkaar verbonden kunnen worden. De achterzijde van een stuk interface ziet er dan uit als in figuur 1. De pennen zijn genummerd op een - op het eerste gezicht nogal ingewikkelde - manier, die echter in het gebruik snel went en die bij digitalici en bedraad(st)ers geheel ingeburgerd is.

De digitalikus maakt aan de hand van zijn schemaas een lijst per konnektor van de voorkomende signalen. De bedraad(st)er heeft liever een lijst per signaal van de pennen waarop dit signaal voorkomt. De brug tussen beiden wordt geslagen door een sorteerprogramma. Het sorteren op de hand zou ondoenlijk zijn, wegens het hoge aantal signalen in een interface (tot enkele duizenden).

Naast het sorteren is er nog het probleem van de volgorde: Moet een aantal pennen elektries verbonden worden, dan kan dat op vele manieren. Een eerste eis is dat per pen hoogstens twee draden bevestigd mogen worden. Dit in verband met eventuele latere wijzigingen, foutkorrekties, en ook i.v.m. de lengte van de pen. De totale draadlengte moet liefst minimaal zijn, om overspraak te vermijden en om de bedrading zo overzichtelijk mogelijk te houden. Zowel het sorteren als het minimaliseren van de draadlengte worden door het programma BAKALG gedaan. Voor het sorteren worden de procedures van A.C. IJsselstein gebruikt[2]. Deze berusten op de rekursieve q-sort algoritme van M.H. van der Emden, en zijn - ondanks dat ze in Algol-60 zijn geschreven - sneller dan de nu gebruikte Shell-sort metode in ELAN. Het lengte-optimalisatie probleem is zo aardig dat we er in het volgende paragraafje iets dieper op ingaan. Het blijkt een standaard probleem te zijn uit de mathematiese besliskunde, waarover veel artikelen gepubliceerd zijn (zie bijv. de referenties in [1]).

3. Het handelsreizigersprobleem

Een handelsreiziger, die woont in een stad 1, wil n-1 andere steden allemaal precies éénmaal bezoeken en daarna naar stad 1 terugkeren. Bovendien wil hij graag zo snel mogelijk weer thuis zijn om 's avonds naar de barend servet show te kijken. In zijn zakagenda staat een afstandstabel waarin alle afstanden tussen twee steden te vinden zijn. Zijn probleem lijkt heel eenvoudig: om zo snel mogelijk thuis te zijn, moet hij zo

weinig mogelijk omwegen maken in zijn roete. Hij schrijft alle mogelijke volgordes op, telt de bijbehorende afstanden bij elkaar op, en kiest de volgorde die de kortste totale afstand oplevert.

Dat het niet zo eenvoudig ligt, blijkt uit een simpele berekening:
vanuit stad 1 heeft hij de keus tussen (n-1) steden
vanuit stad 2 heeft hij de keus tussen (n-2) steden
.....
vanuit stad n-2 heeft hij de keus tussen 2 steden
vanuit stad n-1 moet hij terug naar stad 1

Het totale aantal mogelijke roetes bedraagt dus
$(n-1) \times (n-2) \times \ldots\ldots \times 2 \times 1 = (n-1)!$

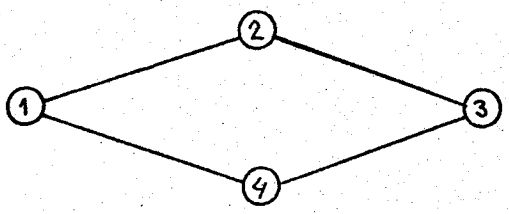
Om welke aantallen het kan gaan zien we in het volgende tabelletje:

aantal steden: n	aantal verschillende roetes: (n-1)!
2	1
3	2
4	6
5	24
10	3628800
20	$\approx 0.24 \times 10^{19}$
30	$\approx 0.27 \times 10^{38}$

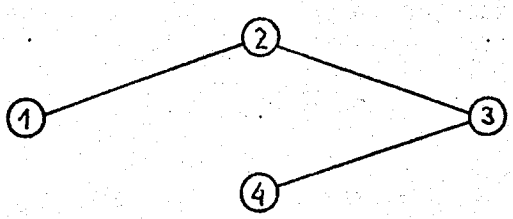
Het is duidelijk, dat onze reiziger, als hij de beste volgorde voor 20 of 30 steden wil weten, 's avonds voor de teevee nog steeds zit te rekenen. Zelfs voor een komputer, die toch wel een miljoen keer sneller rekent, wordt een 30-steden probleem op deze manier wel een onmogelijke zaak. Gelukkig zijn er in de besliskunde metodes ontwikkeld die het handelsreizigersprobleem oplossen zonder alle mogelijke roetes te beschouwen. Een overzicht van deze metodes is o.a. te vinden in [2].
De metodes kunnen onderscheiden worden in eksakte en benaderende metodes. Eksakte metodes vinden gegarandeerd altijd de allerkort-

ste roete, maar de rekentijd neemt minstens exponentieel toe met n.
Met benaderende metodes vindt men soms ook de optimale roete, maar soms ook een roete die bijna optimaal is. Met benaderende metodes neemt de rekentijd veel minder snel toe (bijv. met n^3) en men is in staat om problemen met 500 steden aan te pakken[3]. De metode die wij zullen volgen is de zogenaamde 3-opt metode van Lin[4]. Lin introduceert het begrip "r-optimaal": Een roete heet r-optimaal als het onmogelijk is de roete te verkorten door hem op r plaatsen door te knippen en de resterende r stukken op een andere manier aan elkaar te verbinden. Voor het n-steden probleem is een n-opt roete de optimale, zoals we onmiddellijk inzien. Een r-opt roete, met $1 \leq r < n$, hoeft niet optimaal te zijn. Berekeningen, waarbij veel random problemen eksakt en benaderd werden opgelost, leren dat r=3 het meest ekonomiese kompromis is tussen optimaliteit en rekensnelheid. De rekentijd is dan evenredig met n^3 en de totale lengte wijkt slechts van de orde van 1 procent af van de eksakte oplossing.

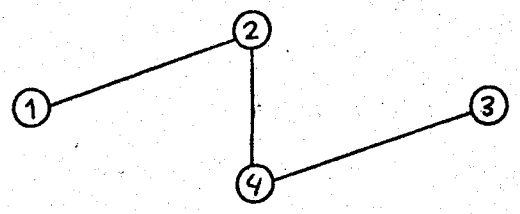
Het Lin-algoritme, zoals BAKALG het gebruikt, levert 3-opt oplossingen voor een gesloten roete langs alle punten, dus inklusief de terugweg naar het uitgangspunt. Om elektries kontakt tussen n pennen te maken, zijn n-1 draadjes al voldoende. Dit wil zeggen dat we van de n verbindingen van de gesloten roete er één (bijvoorbeeld de langste) mogen wegknippen. Dat de overblijvende open roete niet altijd optimaal is blijkt uit een tegenvoorbeeldje van J.K. Lenstra:



De aangegeven roete is optimaal. Weglaten van de kortste verbinding geeft bijvoorbeeld:

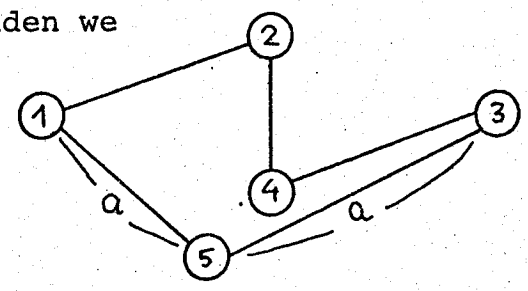


Deze is niet optimaal want de volgende roete is korter:



Om deze moeilijkheid te omzeilen kunnen we gebruik maken van de volgende standaard truuk:
Voer een "virtueel" extra punt n+1 in dat gelijke afstanden a heeft tot alle punten van het probleem. (Dat zo'n punt in het algemeen niet bestaat is onbelangrijk, omdat het algoritme alleen van de onderlinge afstanden uitgaat, die vrij te kiezen zijn. We voeren a.h.w. een extra stad in die per vliegtuig vanuit alle andere steden in dezelfde tijd bereikbaar is).
We zoeken nu een optimale gesloten roete voor het (n+1)-steden probleem. Nu geldt, als we de oplossing nummeren in volgorde 1... (n+1):

$r_{12} + r_{23} + \ldots\ldots + r_{n-1,n} + r_{n,n+1} + r_{n+1,1}$ is minimaal,
maar $r_{n,n+1} = r_{n+1,1} = a$ is een konstante, onafhankelijk van de oplossingsvolgorde 1...n
dus
$r_{12} + r_{23} + \ldots\ldots + r_{n-1,n}$ is minimaal, onafhankelijk van de oplossingsvolgorde 1....n.
En dit is juist de lengte van de open roete langs n punten. In ons voorbeeld vinden we



Als we punt 5 weghalen vinden we dus duidelijk de optimale open roete 1 2 4 3.
Op deze manier worden dus ook begin- en eindpunt van de open roete vrijgelaten in de optimalisatie.

4. Syntax van de invoerfile

1. <inputfile>::= <input><5 nlcr's>
2. <input>::= <element>|<input><element>
3. <element>::= <konnektorkode>(<signaallijst>)<NLCR>
4. <konn.kode>::= <bak><helft><nummer>
5. <bak>:= A|B|C|D|E|F|G
6. <helft>:= A|B
7. <nummer>:= 01|02|03|.....|31|32
8. <signaallijst>:= <signaal>|<signaallijst><signaal>
9. <signaal>:= <pen>=<naam><delimiter>
10. <naam>:= <sim 1><sim 2><sim 3><sim 3><sim 3>|<naam><pitje>
11. <sim 1>:= <letter>|<cijfer>
12. <sim 2>:= <letter>
13. <sim 3>:= <letter>|<cijfer>|<geen simbool>
14. <pitje>:= <simbool'>
15. <delimiter>:= <puntkomma>|<puntkomma><nlcr>
16. <pen>:= <penletter><pencijfer>|<pen><ster>
17. <penletter>:= A|B|C|D|E|F|H|J|K|L|M|N|P|R|S|T|U|V
18. <pencijfer>:= 1|2
19. <ster>:= <simboolx>|<spatie>

Opm: zie als voorbeeld de inputfile uit §6.

5. Gebruiksaanwijzing

Het gebruik van BAKALG is eenvoudig: door op een van de WAMMES time-sharing teleksen het kommando

START BAKALG

te geven, wordt het programma van de lotus-tape gehaald en reageert met

EXECUTION
FILENAAM+

Tiep nu de naam in van een symboliese drumfile, waarin de invoer staat opgeslagen. (Een papierband kan met MONT1 op drum gezet worden). Voldoet de invoer niet aan de syntax uit §4, dan volgen gedetailleerde foutmeldingen over de regeldrukker en BAKALG wordt beëindigd. Met MONT1 kunnen de fouten in de drumfile gekorrigeerd worden, waarna BAKALG opnieuw gestart moet worden (zie boven). Is de invoer foutloos, dan verschijnt na zekere tijd een bedradingslijst op de regeldrukker en als ponsband op de bandponser.

6. Resultaten

In deze paragraaf vinden we een demonstratie voor een test-invoerfile. Rekentijden hangen sterk af van het optimalisatie proses. Voor op het IKO gebruikelijke interfaces ligt de rekentijd tussen 10 en 45 minuten op de EL-X8. Zonder optimalisatie ligt dit tussen 1 en 10 minuten, dus in dezelfde orde als voor BAKSORT. Verder zijn enkele voorbeelden getekend ter vergelijking van resultaten van BAKSORT en BAKALG.

```
 0
 1
 2
 3     AA07(C2 =GRAP.S2 =GRAP.)
 4     AA25(M1 =GRAP.)
 5     AA28(U2 =GRAP.)
 6     AB05(C1 =GRAP.)
 7     AB28(V1 =GRAP.)
 8     AB30(J2 =GRAP.)
 9     BA08(D1 =GRAP.)
10     BA12(U2 =GRAP.)
11     BA17(T1 =GRAP.)
12     BA21(U2 =GRAP.)
13     BA26(C2 =GRAP.)
14     BA30(B1 =GRAP.)
15     BB06(B1 =GRAP.)
16     BB15(B1 =GRAP.)
17     BB18(B1 =GRAP.)
18     BB23(B1 =GRAP.)
20
29
30
31
32
21
22
23
24
25     ....NIET AAN DE PRAKTYK ONTLEEND GEVAL, DAT
26     ....ZONDER OPTIMALISATIE EEN PATOLOGIESE
27     ....BEDRADINGSVOLGORDE OPLEVERT.
```

Voorbeeld van een (kleine) invoerfile.

in het algemeen komen per konnektor meer signaalnamen voor. voor deze demonstratie hebben we ons beperkt tot één signaal. (GRAP).

BEDRADINGSLIJST BAKALG IKO AMSTERDAM BLAD NO% 1

VAN PEN%	NAAR PEN%	LENGTE (CM)%	SIGNAALNAAM%
AA07 C2	AA07 S2	8	GRAP
AB05 C1	BA08 D1	12	
AA07 S2	AB05 C1	12	
BB06 B1	BA12 U2	16	
BA08 D1	BB06 B1	12	
BB15 B1	BA17 T1	8	
BA12 U2	BB15 B1	12	
BB18 B1	BA21 U2	12	
BA17 T1	BB18 B1	8	
BB23 B1	BA26 C2	12	
BA21 U2	BB23 B1	8	
AB28 V1	BA30 B1	8	
BA26 C2	AB28 V1	8	
AB30 J2	AA28 U2	12	
BA30 B1	AB30 J2	12	
AA28 U2	AA25 M1	12	

VOOR DEZE BAK ZYN NODIG%

6 DRAADJES VAN 8 CENTIMETER
9 DRAADJES VAN 12 CENTIMETER
1 DRAADJES VAN 16 CENTIMETER

HET TOTALE AANTAL DRAADJES IS% 16

DEZE BEDRADINGSLIJST IS GEMAAKT DOOR HET
BEDRADINGSPROGRAMMA BAKALG,DAT DE DRAADLENGTE
MINIMALISEERT MET HET R-OPT ALGORITME VAN
-LIN.

HET AANTAL BEDRADE PENNEN IN DEZE BAK IS% 17

DAT WAS HET DAN WEER
VOOR DEZE KEER....
HOEWEL....
730302 17 UUR 50
16 MILLI-UUR REKENTYD

Bedradingslijst bij voorgaand voorbeeld.

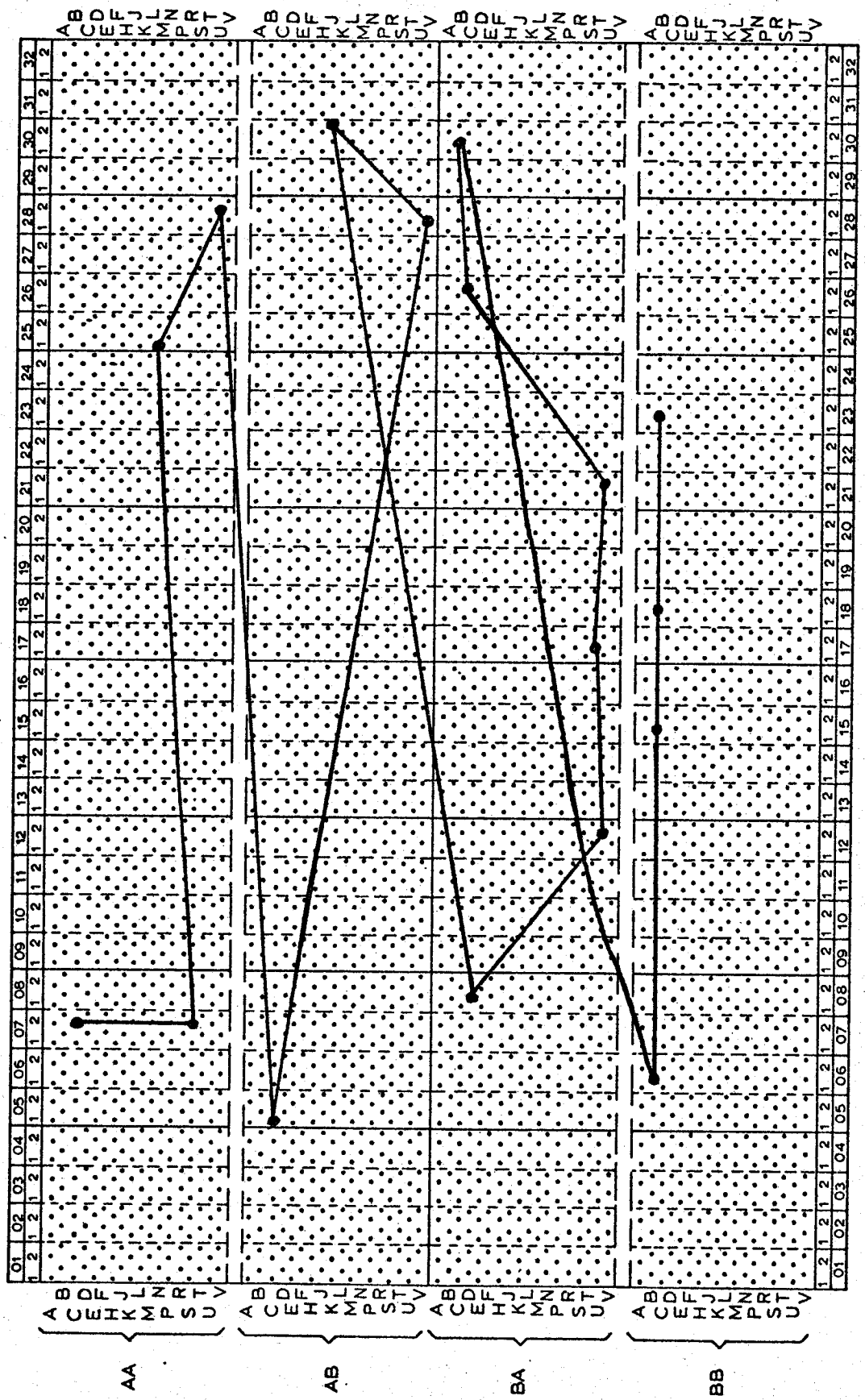


fig.1. bedrading zonder optimalisatie (BAKSORT).
voor het signaal GRAP uit het voorbeeld.

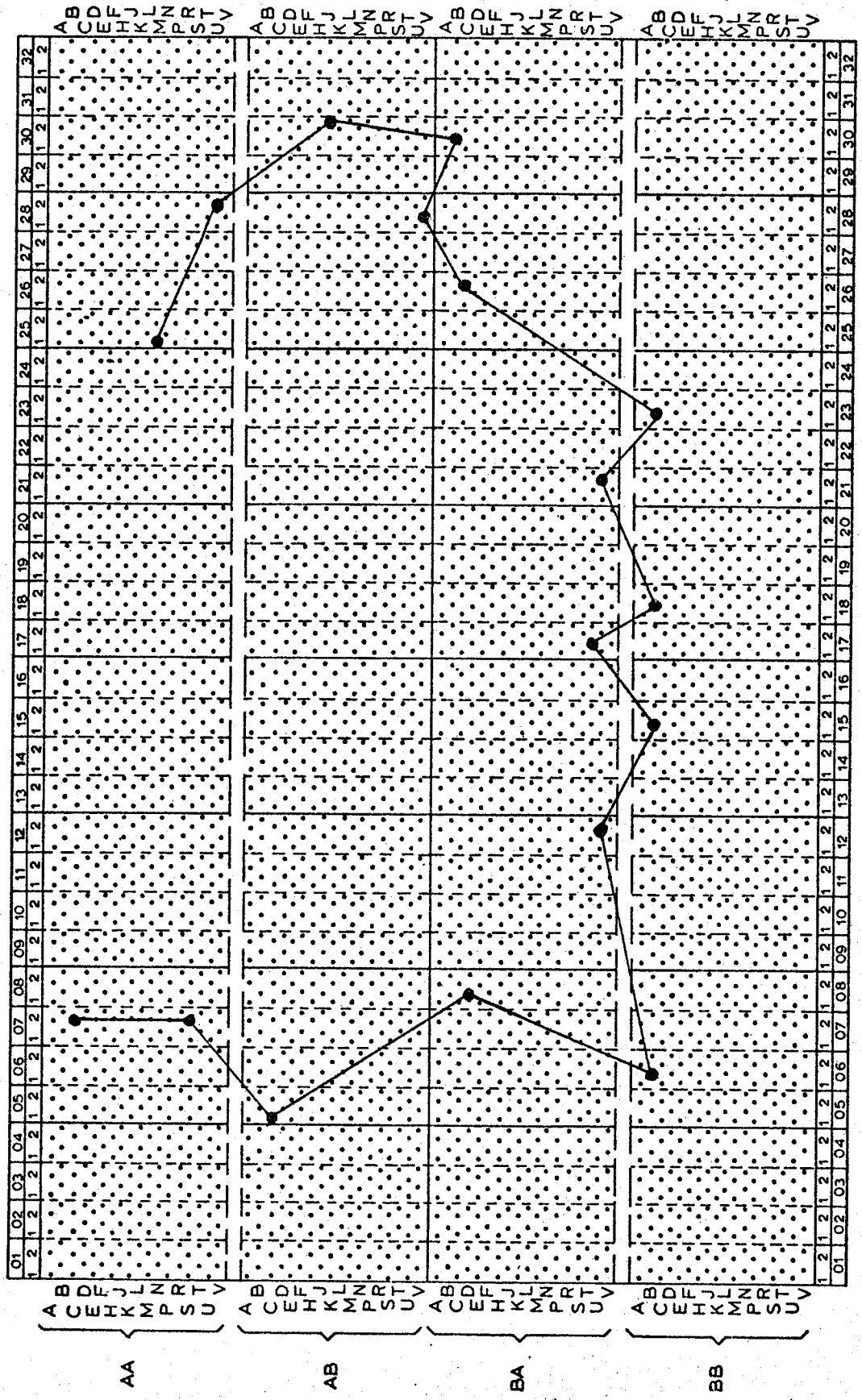


fig.2. bedrading van GRAP met optimalisatie (BAKALG).

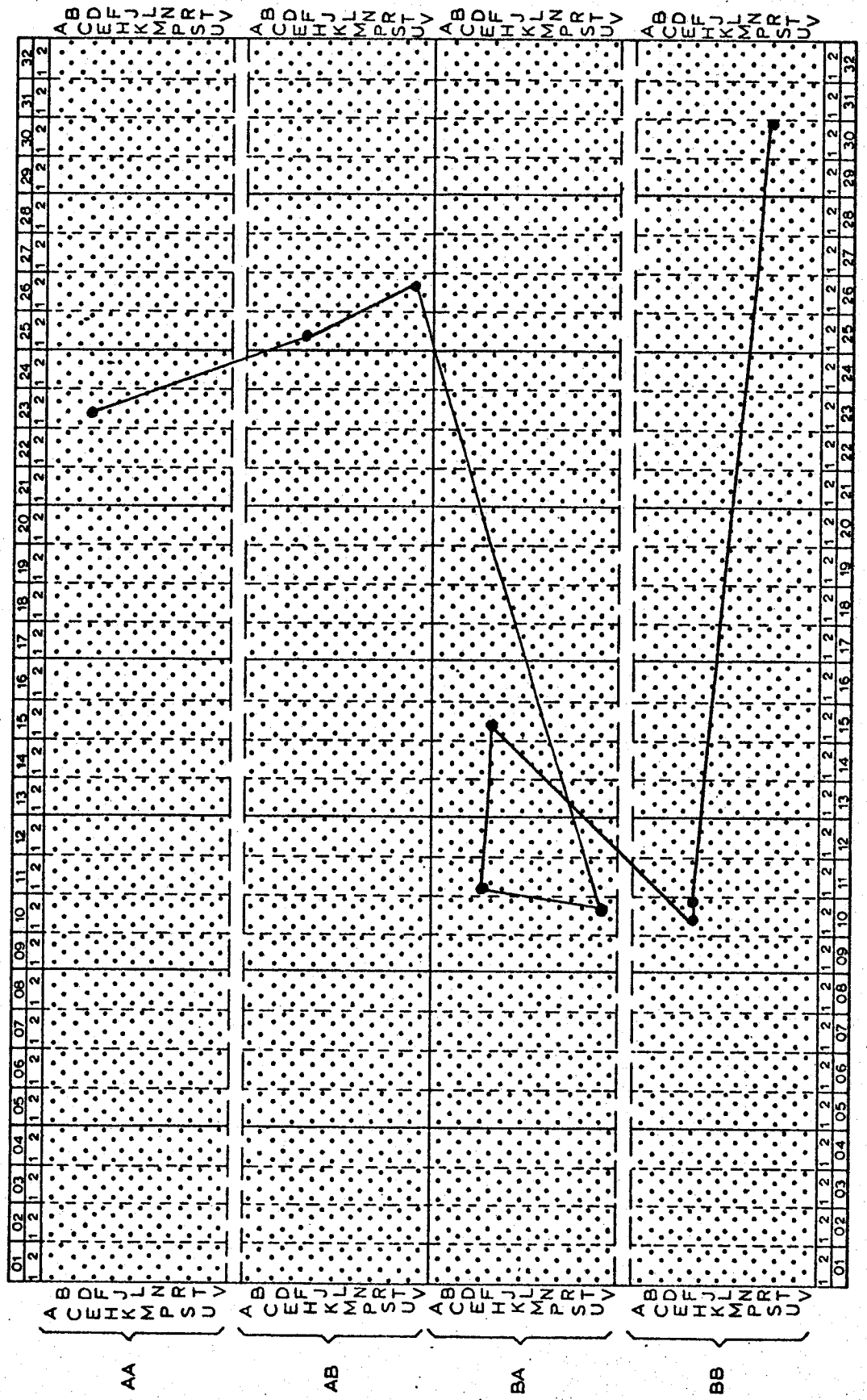


fig 3. een praktijkgeval zonder optimalisatie.

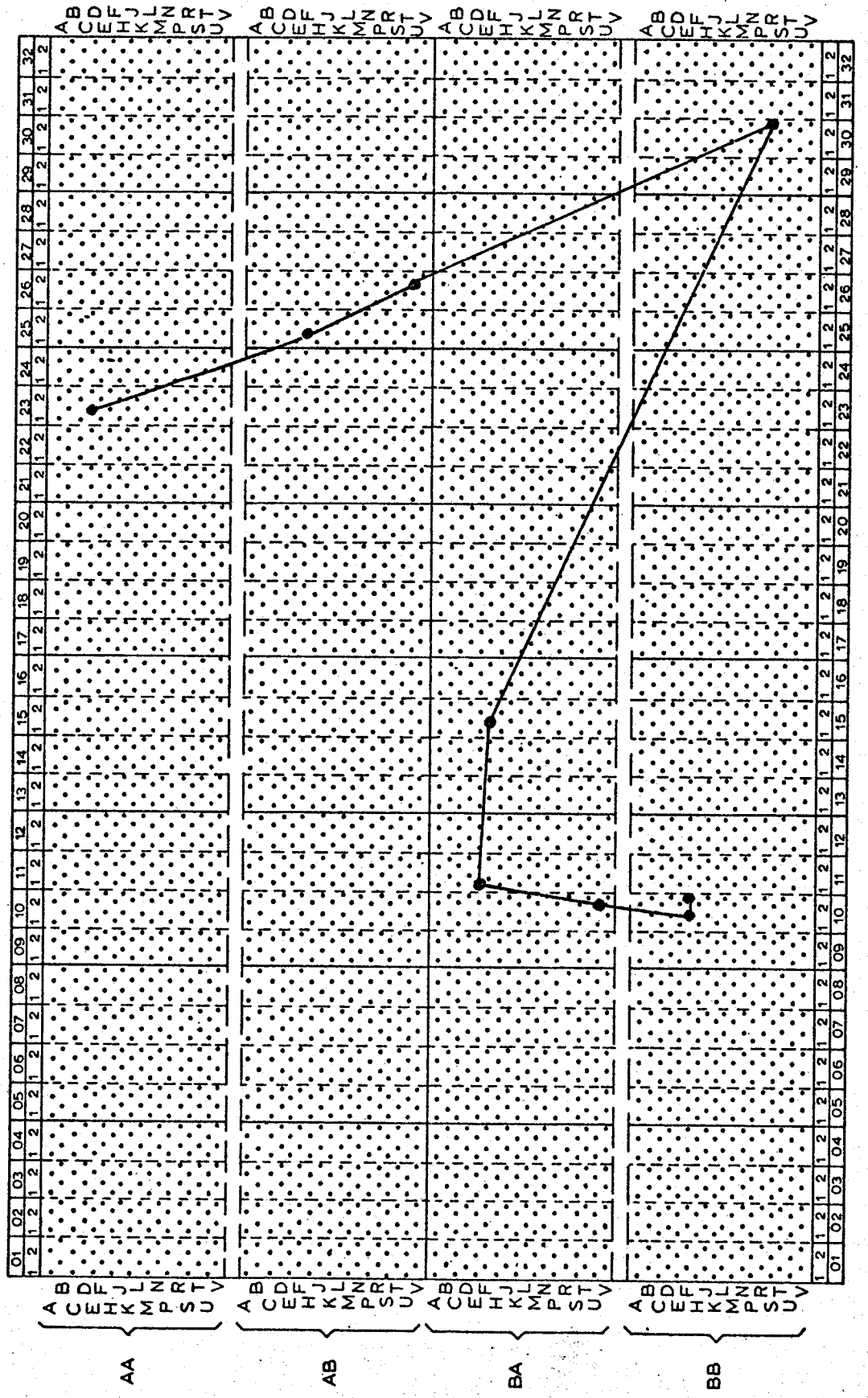


fig.4. als fig.3 maar met optimalisatie.

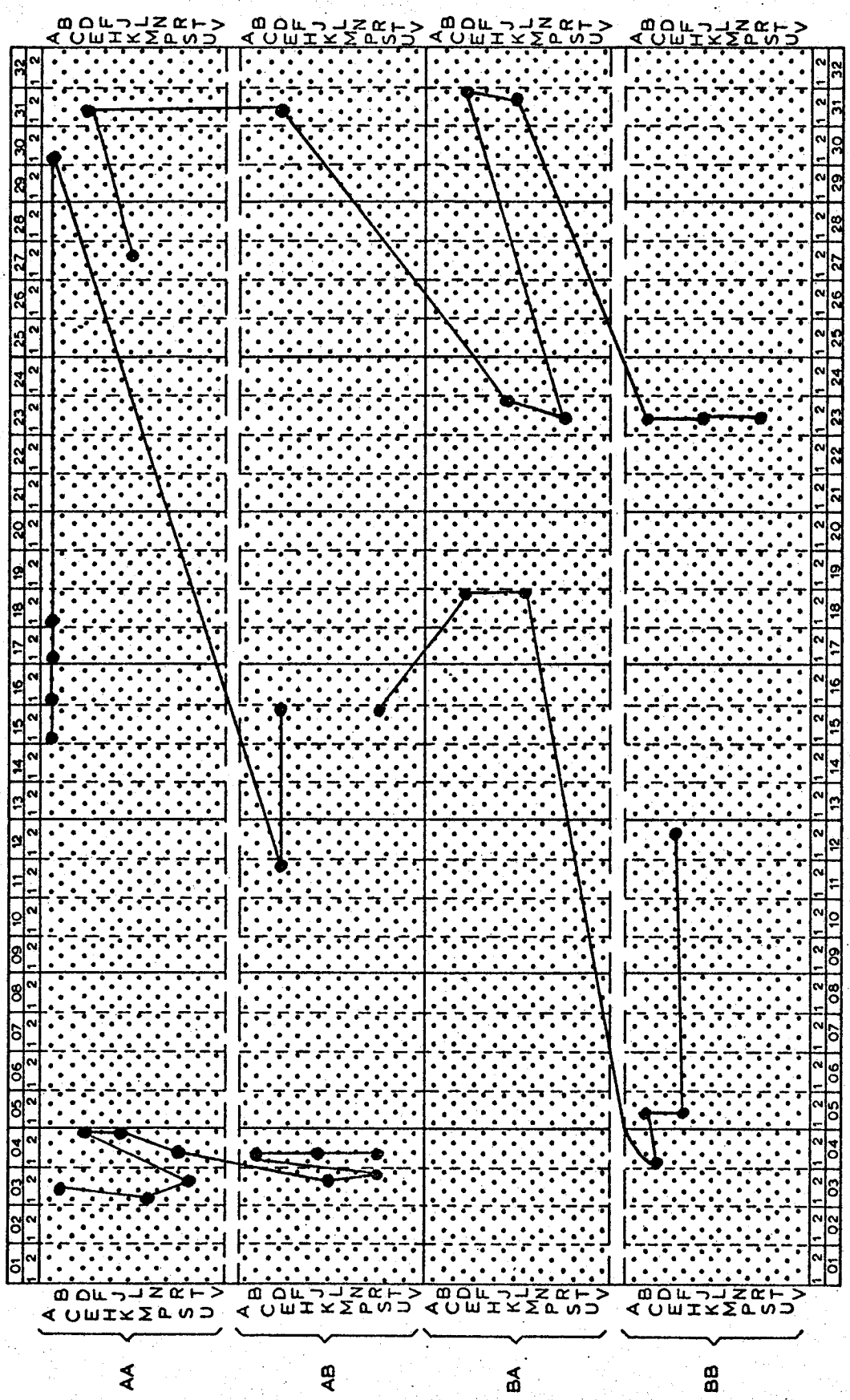


fig.5. nog enkele praktijkgevallen.

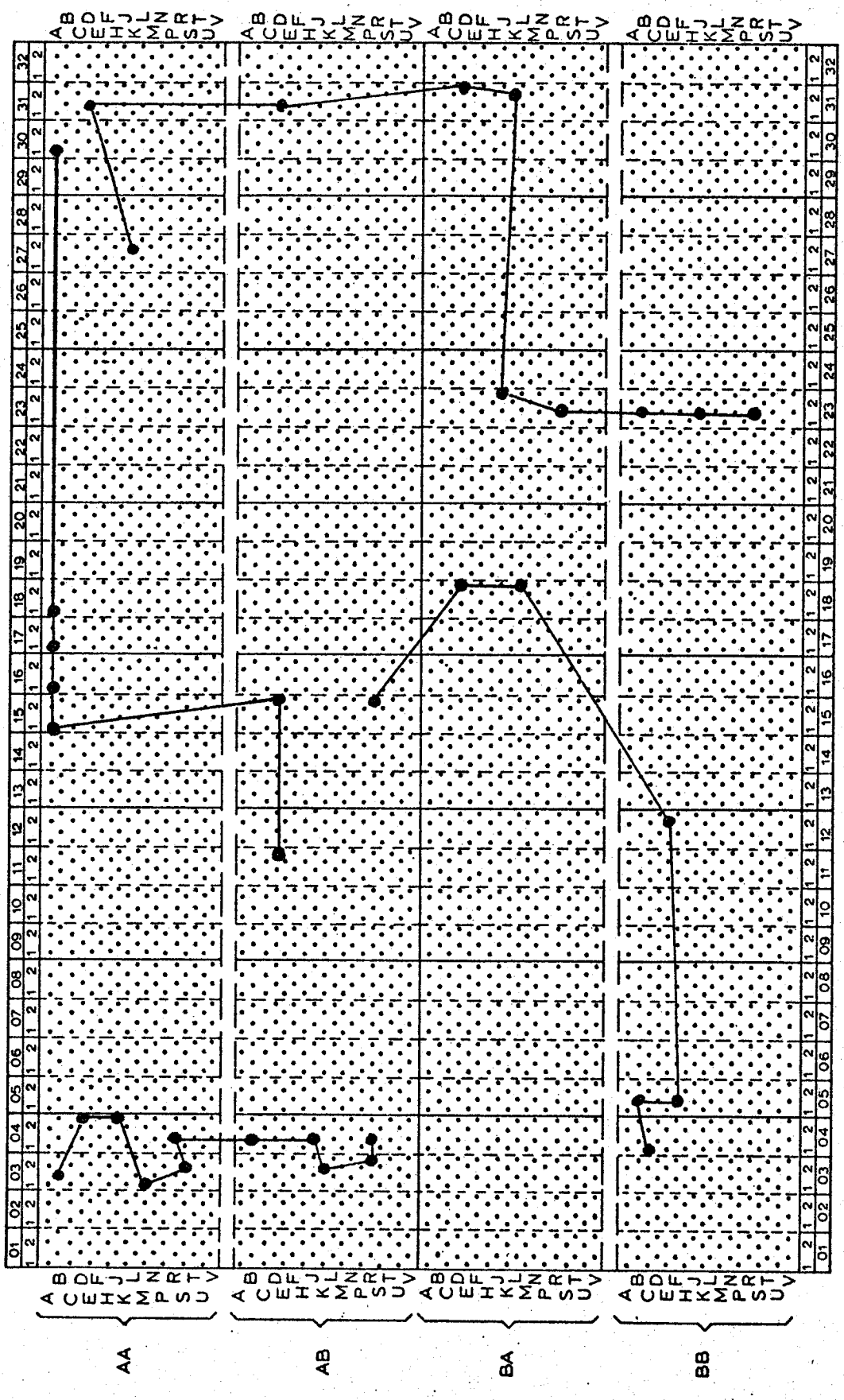


fig.6. als fig.5 maar met optimalisatie.

7. Literatuur

1. A.C. IJsselstein, "Voorlopige beschrijving sorteerprocedures", (Math. Centrum A'dam, 25-7-72)

2. J.K. Lenstra, "Branch and Bound Algorithmen voor het handelsreizigersprobleem", (Math. Centrum A'dam, BN 16/72,voorlopige uitgave)

3. Nicos Christofides and Samuel Eilon, "Algorithms for large-scale travelling salesman problems", Operat. Research Quarterly 23,4,(1972)

4. S. Lin, "Computer Solutions of the Travelling Salesman problem", Bell Systems Techn. Journal 44,2245(1965)

```
 1
 2 'BEGIN'  'COMMENT'
 3 BAKALG, BEDRADINGSPROGRAMMA VOOR INTERFAECES,MET
 4 ZOEKPROCEDURE OM DE KORTSTE DRAADLENGTE VOOR EEN
 5 AANTAL VERBINDINGEN BIJ EEN SIGNAAL TE BEREIKEN.
 6
 7 'INTEGER'POINTER,SYM1,SYM2,SYM3,SYM4,SYM5,REGELNR.
 8 'INTEGER'REGEL,BLAD.
 9 'BOOLEAN' FOUTJE.
10 'INTEGER''ARRAY' LYST1,LYST2(0%4000).
11 'INTEGER''ARRAY'DRAADJE(2%50).
12
13                          'PROCEDURE' LEES.
14 'BEGIN''COMMENT' LEEST 5 SIMBOLEN VOORUIT,BIJ
15 5 NLCR'S WORDT AANGENOMEN DAT DE INVOER UIT-
16 GEPUT IS.
17 NOGES%
18 SYM1%=SYM2.SYM2%=SYM3.SYM3%=SYM4.SYM4%=SYM5.
19 SYM5%=RESYM.
20 'IF'SYM5'GT'36'AND'SYM5'LT'63'THEN'
21 SYM5%=SYM5-27.
22 'IF'SYM5=119'THEN'REGELNR%=REGELNR+1.
23 'IF'(SYM1=119'AND'SYM2=119'AND'
24 SYM3=119'AND'SYM4=119'AND'
25 SYM5=119)'THEN'
26 'GOTO' EINDEBAND.
27 'END'  LEES.
28
29                          'PROCEDURE'FOUT(A).
30 'STRING'A.'BEGIN'NLCR.
31 PRINTTEXT(A).TAB.PRINTTEXT('(' IN REGEL')').
32 ABSFIXT(5,0,REGELNR).TAB.
33 PRINTTEXT('('LAATSTE GOEDE SIGNAAL=')').
34 PRINTNAAM(LYST1(POINTER-1),'FALSE').
35 FOUTJE%='TRUE'.
36 FF%'IF''NOT'HAAKSLUIT(SYM1)'THEN''BEGIN'
37 LEES.'GOTO'FF'END''ELSE''GOTO'BEGINOFPROG.
38 'END'   FOUT.
39
40 'BOOLEAN'                          'PROCEDURE'SPATIE(SYM).
41 'INTEGER'SYM.
42 'IF'SYM=93'THEN'SPATIE%='TRUE'
43 'ELSE'SPATIE%='FALSE'.
44
45 'BOOLEAN'                          'PROCEDURE'LETTER(SYM).
46 'INTEGER'SYM.
47 'IF'SYM'GT'9'AND'SYM'LT'36'THEN'LETTER%='TRUE'
48 'ELSE'LETTER%='FALSE'.
49
50 'BOOLEAN'                          'PROCEDURE'CYFER(SYM).
51 'INTEGER'SYM.
52 'IF'SYM'LT'10'THEN'
53 CYFER%='TRUE''ELSE'CYFER%='FALSE'.
54
55 'BOOLEAN'                          'PROCEDURE'PITJE(SYM).
56 'INTEGER'SYM.
57 'IF'SYM=120'THEN'PITJE%='TRUE''ELSE'PITJE%='FALSE'.
58
59 'BOOLEAN'                          'PROCEDURE'GELYK(SYM).
60 'INTEGER'SYM.
```

```
'IF'SYM=70'THEN'GELYK%='TRUE''ELSE'GELYK%='FALSE'.

'BOOLEAN'                              'PROCEDURE'PUNTKOMMA(SYM).
'INTEGER'SYM.
'IF'SYM=91'THEN'PUNTKOMMA%='TRUE'
'ELSE'PUNTKOMMA%='FALSE'.

'BOOLEAN'                              'PROCEDURE' HAAKOPEN(SYM).
'INTEGER'SYM.
'IF'SYM=98'THEN'HAAKOPEN%='TRUE'
'ELSE'HAAKOPEN%='FALSE'.

'BOOLEAN'                              'PROCEDURE'HAAKSLUIT(SYM).
'INTEGER'SYM.
'IF'SYM=99'THEN'HAAKSLUIT%='TRUE'
'ELSE'HAAKSLUIT%='FALSE'.

'BOOLEAN'                              'PROCEDURE'STER(SYM).
'INTEGER'SYM.
'IF'SYM=66'THEN'STER%='TRUE'
'ELSE'STER%='FALSE'.

                          'PROCEDURE'PRINTNAAM(N,P).
'VALUE' N.
'INTEGER' N.
'BOOLEAN'P.
'BEGIN''INTEGER'A,I,DELER.
'INTEGER''ARRAY'S(1%6).
'FOR'I%=5'STEP'-1'UNTIL'2'DO''BEGIN'
A%=N.
DELER%='IF'I=2'THEN'27'ELSE'37.
N%=N'DIV'DELER.
S(I)%=A-N*DELER     'END'.
S(1)%=N.
'FOR'I%=1'STEP'1'UNTIL'5'DO'
'IF'S(I)'NE'36'THEN'PRSYM
('IF'I'NE'2'THEN'S(I)'ELSE'S(I)+10).
'IF'P'THEN'PRSYM(120).
'END'  PRINTNAAM.


                          'PROCEDURE'PRINTKONPEN(X,Y,S).
'VALUE'X,Y.
'INTEGER'X,Y.'BOOLEAN'S.
'BEGIN''INTEGER'BAK,HELFT,KON1,KON2.
BAK%=Y'DIV'42.
HELFT%=(Y-BAK*42)'DIV'22.
X%=X+4.
KON1%=X'DIV'40.
KON2%=X'DIV'4-KON1*10.
X%=X-4.
PRSYM(BAK+10).PRSYM(HELFT+10).
PRSYM(KON1).PRSYM(KON2).SPACE(1).
Y%=Y-BAK*42-HELFT*22.
X%=X-(X'DIV'4)*4.
PRSYM(
'IF'Y=0'THEN'10'ELSE'
'IF'Y=1'THEN'11'ELSE'
'IF'Y=2'THEN'12'ELSE'
'IF'Y=3'THEN'13'ELSE'
```

```
121 'IF'Y=4'THEN'14'ELSE'
122 'IF'Y=5'THEN'15'ELSE'
123 'IF'Y=6'THEN'17'ELSE'
124 'IF'Y=7'THEN'19'ELSE'
125 'IF'Y=8'THEN'20'ELSE'
126 'IF'Y=9'THEN'21'ELSE'
127 'IF'Y=10'THEN'22'ELSE'
128 'IF'Y=11'THEN'23'ELSE'
129 'IF'Y=12'THEN'25'ELSE'
130 'IF'Y=13'THEN'27'ELSE'
131 'IF'Y=14'THEN'28'ELSE'
132 'IF'Y=15'THEN'29'ELSE'
133 'IF'Y=16'THEN'30'ELSE'
134 31);
135 'IF'X=0'OR'X=1'THEN'PRSYM(1)'ELSE'
136 'IF'X=2'OR'X=3'THEN'PRSYM(2).
137 'IF'S'THEN'PRSYM(66).
138 'END'PRINTKONPEN.
139
140                          'PROCEDURE'DEKODEER(W,X,Y,P,S).
141 'INTEGER'W,X,Y.'BOOLEAN'P,S.
142 'BEGIN'
143 X%=LYST2(W).'IF'X'GE'100000'THEN''BEGIN'
144 P%='TRUE'.X%=X-100000'END'
145 'ELSE'P%='FALSE'.
146 Y%=X'DIV'2.
147 'IF'X-2*Y'GT'0'THEN''BEGIN'
148 S%='TRUE'.X%=X-1'END'
149 'ELSE'S%='FALSE'.
150 Y%=X'DIV'500.
151 X%=(X-Y*500)'DIV'2.
152 'END'  DEKODEER.
153
154
155
156                          'PROCEDURE'LEESNAAM(N,P).
157 'INTEGER'N.'BOOLEAN'P.
158 'BEGIN'
159 'INTEGER'I.
160 LEES.
161 'IF''NOT'GELYK(SYM1)'THEN'
162 FOUT('('VOOR SIGNAALNAAM GEEN =')').
163 'IF''NOT'(LETTER(SYM2)'OR'CYFER(SYM2))'THEN'
164 FOUT('('SIGN.NAAMBEGINT NIET MET LETTER OF CYFER')').
165 'IF''NOT'LETTER(SYM3)'THEN'
166 FOUT('('SIGNAALNAAM HEEFT GEEN LETTER OP 2E PLAATS')').
167 N%=0.
168 'FOR'I%=1'STEP'1'UNTIL'5'DO'
169 'BEGIN'
170 'IF'LETTER(SYM2)'OR'CYFER(SYM2)'THEN'LEES
171 'ELSE''IF'PITJE(SYM2)'OR'PUNTKOMMA(SYM2)
172 'THEN'SYM1%=36'ELSE'
173 FOUT('('SIGNAALNAAM BEVAT FOUT SIMBOOL')').
174 N%=N*
175 ('IF'I=2'THEN'27'ELSE'37)+
176 ('IF'I=2'THEN'SYM1-10'ELSE'SYM1)
177 'END'.
178 'IF'PITJE(SYM2)'THEN''BEGIN'
179 LEES.P%='TRUE''END''ELSE'P%='FALSE'.
180 LEES.
```

```
181 'IF''NOT'PUNTKOMMA(SYM1)'THEN'
182 FOUT('('SIGNAALNAAM EINDIGT NIET OP . ')').
183 'END'   LEESNAAM.
184
185
186                              'PROCEDURE'VEC2QSORT(A1,A2,L1,U1).
187 'VALUE'L1,U1.
188 'INTEGER'L1,U1.
189 'ARRAY'A1,A2.
190 'BEGIN''COMMENT' VEC2QSORT SORTEERT DE ELEMENTEN
191 VAN TWEE EENDIMENSIONALE ARRAYS OP VOLGORDE VAN GROOTTE
192 VAN DE ELEMENTEN VAN HETN EERSTE ARRAY.
193 'INTEGER'P,Q,IX,IZ.
194 'REAL'X,XX,Y,ZZ,Z.
195                              'PROCEDURE'VEC2SORT.
196 'BEGIN''INTEGER'L,U.
197 L%=L1.U%=U1.
198 PART%
199 P%=L.Q%=U.X%=A1(P).Z%=A1(Q).
200 'IF'X'GT'Z'THEN'
201 'BEGIN'Y%=X.A1(P)%=X%=Z.A1(Q)%=Z%=Y.
202 Y%=A2(P).A2(P)%=A2(Q).A2(Q)%=Y 'END'.
203 'IF'U-L'GT'1'THEN'
204 'BEGIN'XX%=X.IX%=P.ZZ%=Z.IZ%=Q.
205 LEFT%
206 'FOR'P%=P+1'WHILE'P'LT'Q'DO''BEGIN'
207 X%=A1(P).'IF'X'GE'XX'THEN''GOTO'RIGHT'END'.
208 P%=Q-1.'GOTO'OUT.
209 RIGHT%
210 'FOR'Q%=Q-1'WHILE'Q'GT'P'DO''BEGIN'
211 Z%=A1(Q).
212 'IF'Z'LE'ZZ'THEN''GOTO'DIST'END'.
213 Q%=P.P%=P-1.Z%=X.X%=A1(P).
214 DIST%
215 'IF'X'GT'Z'THEN''BEGIN'
216 Y%=X.A1(P)%=X%=Z.A1(Q)%=Z%=Y.
217 Y%=A2(P).A2(P)%=A2(Q).A2(Q)%=Y'END'.
218 'IF'X'GT'XX'THEN''BEGIN'XX%=X.IX%=P'END'.
219 'IF'Z'LT'ZZ'THEN''BEGIN'ZZ%=Z.IZ%=Q'END'.
220 'GOTO'LEFT.
221 OUT%
222 'IF'P'NE'IX'AND'X'NE'XX'THEN''BEGIN'
223 A1(P)%=XX.A1(IX)%=X.
224 Y%=A2(P).A2(P)%=A2(IX).A2(IX)%=Y'END'.
225 'IF'Q'NE'IZ'AND'Z'NE'ZZ'THEN''BEGIN'
226 A1(Q)%=ZZ.A1(IZ)%=Z.
227 Y%=A2(Q).A2(Q)%=A2(IZ).A2(IZ)%=Y'END'.
228 'IF'U-Q'GT'P-L'THEN''BEGIN'L1%=L.U1%=P-1.L%=Q+1
229 'END''ELSE''BEGIN'U1%=U.L1%=Q+1.U%=P-1'END'.
230 'IF'U1'GT'L1'THEN'VEC2SORT.
231 'IF'U'GT'L'THEN''GOTO'PART.
232 'END'
233 'END'VEC2SORT.
234 'IF'U1'GT'L1'THEN'VEC2SORT
235 'END'VEC2QSORT.
236
237
238
239                              'PROCEDURE' VECQSORT(A,L1,U1).
240 'VALUE'L1,U1.'INTEGER'L1,U1.
```

```
241 'ARRAY'A.
242 'BEGIN''INTEGER'P,Q,IX,IZ.
243 'REAL'X,XX,Y,ZZ,Z.
244                          'PROCEDURE'VECSORT.
245 'BEGIN''INTEGER'L,U.
246 L%=L1.U%=U1.
247 PART%
248 P%=L.Q%=U.X%=A(P).Z%=A(Q).
249 'IF'X'GT'Z'THEN'
250 'BEGIN'Y%=X.A(P)%=X%=Z.
251 A(Q)%=Z%=Y 'END'.
252 'IF'U-L'GT'1'THEN'
253 'BEGIN'XX%=X.IX%=P.ZZ%=Z.IZ%=Q.
254 LEFT%
255 'FOR'P%=P+1'WHILE'P'LT'Q'DO'
256 'BEGIN'X%=A(P).
257 'IF'X'GE'XX'THEN''GOTO' RIGHT 'END'.
258 P%=Q-1.'GOTO' OUT.
259 RIGHT%
260 'FOR'Q%=Q-1'WHILE'Q'GT'P'DO'
261 'BEGIN'Z%=A(Q).'IF'Z'LE'ZZ'THEN''GOTO' DIST 'END'.
262 Q%=P.P%=P-1.Z%=X.X%=A(P).
263 DIST%
264 'IF'X'GT'Z'THEN''BEGIN'
265 Y%=X.A(P)%=X%=Z.A(Q)%=Z%=Y'END'.
266 'IF'X'GT'XX'THEN''BEGIN'XX%=X.IX%=P'END'.
267 'IF'Z'LT'ZZ'THEN''BEGIN'ZZ%=Z.IZ%=Q'END'.
268 'GOTO' LEFT.
269 OUT%'IF'P'NE'IX'AND'X'NE'XX'THEN'
270 'BEGIN'A(P)%=XX.A(IX)%=X'END'.
271 'IF'Q'NE'IZ'AND'Z'NE'ZZ'THEN'
272 'BEGIN'A(Q)%=ZZ.A(IZ)%=Z'END'.
273 'IF'U-Q'GT'P-L'THEN'
274 'BEGIN'L1%=L.U1%=P-1.L%=Q+1'END''ELSE'
275 'BEGIN'U1%=U.L1%=Q+1.U%=P-1'END'.
276 'IF'U1'GT'L1'THEN'VECSORT.
277 'IF'U'GT'L'THEN''GOTO'PART 'END'
278 'END' VECSORT.
279 'IF'U1'GT'L1'THEN' VECSORT
280 'END' VECQSORT.
281
282                          'PROCEDURE'LEESKON(X,Y).
283 'INTEGER'X,Y.
284 'BEGIN''INTEGER'GETAL.
285 X%=Y%=0.
286 LEES.
287 'IF''NOT'LETTER(SYM1)'THEN'
288 FOUT('('BAKKODE BEGINT NIET MET LETTER')').
289 SYM1%=SYM1-10.
290 'IF'SYM1'GT'6'THEN'
291 FOUT('('MEERDAN 7 BAKKEN')').
292 Y%=Y+42*SYM1.
293 LEES.
294 'IF'SYM1=10'THEN''ELSE'
295 'IF'SYM1=11'THEN'Y%=Y+22'ELSE'
296 FOUT('('BAKKODE HEEFT GEEN A OF B OP 2E PLAATS')').
297 LEES.
298 'IF''NOT'CYFER(SYM1)'AND'CYFER(SYM2)'THEN'
299 FOUT('('KONNEKTORNUMMER IS GEEN GETAL')').
300 GETAL%=10*SYM1+SYM2-1.
```

```
'IF'GETAL'GT'31'THEN'
FOUT('('NIET BESTAANDE KONNEKTOR')').
X%=X+4*GETAL.
LEES.LEES.
'IF''NOT'HAAKOPEN(SYM1)'THEN'
FOUT('('NA KONNEKTORKODE GEEN ( ')').
'END'   LEESKON.


                              'PROCEDURE'LEESPEN(X,Y,S).
'INTEGER'X,Y.
'BOOLEAN'S.
'BEGIN'
LEES.
'IF''NOT'LETTER(SYM1)'THEN'
FOUT('('PENKODE BEGINT NIET MET LETTER')').
'IF'SYM1=10'THEN''BEGIN'X%=0.Y%=0'END''ELSE'
'IF'SYM1=11'THEN''BEGIN'X%=1.Y%=1'END''ELSE'
'IF'SYM1=12'THEN''BEGIN'X%=0.Y%=2'END''ELSE'
'IF'SYM1=13'THEN''BEGIN'X%=1.Y%=3'END''ELSE'
'IF'SYM1=14'THEN''BEGIN'X%=0.Y%=4'END''ELSE'
'IF'SYM1=15'THEN''BEGIN'X%=1.Y%=5'END''ELSE'
'IF'SYM1=17'THEN''BEGIN'X%=0.Y%=6'END''ELSE'
'IF'SYM1=19'THEN''BEGIN'X%=1.Y%=7'END''ELSE'
'IF'SYM1=20'THEN''BEGIN'X%=0.Y%=8'END''ELSE'
'IF'SYM1=21'THEN''BEGIN'X%=1.Y%=9'END''ELSE'
'IF'SYM1=22'THEN''BEGIN'X%=0.Y%=10'END''ELSE'
'IF'SYM1=23'THEN''BEGIN'X%=1.Y%=11'END''ELSE'
'IF'SYM1=25'THEN''BEGIN'X%=0.Y%=12'END''ELSE'
'IF'SYM1=27'THEN''BEGIN'X%=1.Y%=13'END''ELSE'
'IF'SYM1=28'THEN''BEGIN'X%=0.Y%=14'END''ELSE'
'IF'SYM1=29'THEN''BEGIN'X%=1.Y%=15'END''ELSE'
'IF'SYM1=30'THEN''BEGIN'X%=0.Y%=16'END''ELSE'
'IF'SYM1=31'THEN''BEGIN'X%=1.Y%=17'END''ELSE'
FOUT('('PENKODE BEVAT VERKEERDE LETTER')').
LEES.
'IF'SYM1=1'THEN''ELSE'
'IF'SYM1=2'THEN'X%=X+2'ELSE'
FOUT('('PENKODE HEEFT GEEN 1 OF 2 ALS 2E SIMBOOL')').
'IF'STER(SYM2)'OR'SPATIE(SYM2)'THEN'
LEES'ELSE'
FOUT('('PENKODE EINDIGT NIET OP SPATIE OF *')').
'IF'STER(SYM1)'THEN'S%='TRUE''ELSE'S%='FALSE'.
'END'    LEESPEN.

'REAL'                              'PROCEDURE'AFSTAND(W1,W2).
'INTEGER'W1,W2.
'BEGIN''INTEGER'X1,X2,Y1,Y2,X,Y.
'BOOLEAN'P,S.
DEKODEER(W1,X1,Y1,P,S).
DEKODEER(W2,X2,Y2,P,S).
X%=X1-X2.Y%=Y1-Y2.
AFSTAND%=SQRT(100*X*X+100*Y*Y)
'END'   AFSTAND.


                                 'PROCEDURE'BANDLEZEN.
'BEGIN''INTEGER'XK,YK,XP,YP,N.
'BOOLEAN'S,P.
POINTER%=0.
```

```
361 'FOR'N%=1,N+1'WHILE''NOT'LETTER(SYM2)'DO'LEES.
362 NIEUWEKON%
363 LEESKON(XK,YK).
364 NIEUWEPEN%
365 LEESPEN(XP,YP,S).
366 LEESNAAM(N,P).
367 'IF'SYM2=119'THEN'LEES.
368 LYST1(POINTER)%=N.
369 LYST2(POINTER)%=(YK+YP)*500+
370 (XK+XP)*2+('IF'S'THEN'1'ELSE'0)+
371 ('IF'P'THEN'100000'ELSE'0).
372 POINTER%=POINTER+1.
373 'IF''NOT'HAAKSLUIT(SYM2)'THEN''GOTO'NIEUWEPEN.
374 LEES.LEES.
375 'IF''NOT'SYM1=119'THEN'
376 FOUT('('NA ) KOMT GEEN NLCR')').
377 MEER%'IF'SYM2=119'THEN''BEGIN'
378 LEES.'GOTO'MEER'END'.
379 'GOTO'NIEUWEKON.
380 'END'   BANDLEZEN.
381
382                              'PROCEDURE' LIN(Q,N,C,TU,GU).
383 'ARRAY'C,TU,GU.
384 'INTEGER'Q,N.
385 'BEGIN''COMMENT'  LIN ZOEKT Q
386 3-OPTIMALE TOERS LANGS N PUNTEN,WAARVAN
387 DE AFSTANDEN ZIJ N GEGEVEN IN ARRAY C.
388 DE TOERS KOMEN ALS RIJEN INDICES IN
389 ARRAY TU(1%Q,1%N) , DE LENGTES VAN DE
390 TOEREN KOMEN IN G(1%Q).HET PROBLEEM IS
391 IN DE LITTERATUUR BEKEND ALS HET
392 TRAVELING SALESMA PROLEM(TSP).
393 MET DANK AAN HR LENSTRA(MATH SENTRUM).
394 'BEGIN''COMMENT' RANDOM PROCEDURES
395 PSEUDO RANDOM GENERATOR VOLGENS
396 M.GREENBERGER,M.T.A.C.15,1961,383.
397 'INTEGER'XN,LA,MU,P.
398                              'PROCEDURE'SETRANDOM(X).'INTEGER'X.
399 'BEGIN'LA%=25.MU%=3.
400 P%=16*16*16*16*16.
401 XN%=P*X.
402 'END'   SETRANDOM.
403 'REAL'                              'PROCEDURE' RANDOM.
404 'BEGIN'XN%=XN*LA+MU.
405 XN%=XN-(XN'DIV'P)*P.
406 RANDOM%=XN/P'END' RANDOM.
407 'BEGIN''INTEGER'G,H,K,L,IQ,
408 N3,N1,T1,TN,TK,TKK,TL,TLL,CJ,CK,
409 CL,C1,C2,CKKN,I,J.
410 'BOOLEAN'SF.
411 'INTEGER''ARRAY'T,TT(1%N).
412 'INTEGER''ARRAY'S(1%N,1%N).
413 SETRANDOM(0.5).
414 'FOR'I%=1'STEP'1'UNTIL'N'DO'
415 'FOR'J%=1'STEP'1'UNTIL'I'DO'
416 S(I,J)%=S(J,I)%=0.
417 N3%=N-3.N1%=N-1.
418 'FOR'IQ%=1'STEP'1'UNTIL'Q'DO'
419 'BEGIN'SF%=IQ'GT'1.T(1)%=1.
420 'FOR'I%=2'STEP'1'UNTIL'N'DO'
```

```
421 'BEGIN'H%=RANDOM*I+1.5.
422 'FOR'J%=I'STEP'-1'UNTIL'H'DO'
423 T(J)%=T(J-1).
424 T(H-1)%=I.
425 'END'H%=RANDOM.
426 A%
427 'FOR'J%=1'STEP'1'UNTIL'N'DO'
428 'BEGIN'T1%=T(1).TN%=T(N).
429 'IF'SF'AND'S(T1,TN)=1'THEN''GOTO'ROTATION.
430 CJ%=C(T1,TN).
431 'FOR'K%=1'STEP'1'UNTIL'N3'DO'
432 'BEGIN'
433 TK%=T(K).TKK%=T(K+1).
434 CK%=CJ+C(TK,TKK).
435 CKKN%=C(TKK,TN).
436 'FOR'L%=K+1'STEP'1'UNTIL'N1'DO'
437 'BEGIN'
438 TL%=T(L).TLL%=T(L+1).
439 CL%=CK+C(TL,TLL).
440 C1%=C(T1,TL)+C(TK,TLL)+CKKN.
441 C2%=C(T1,TLL)+C(TK,TL)+CKKN.
442 'IF'C1'LT'CL'OR'C2'LT'CL'THEN'
443 'BEGIN'
444 'FOR'I%=1'STEP'1'UNTIL'N'DO'
445 TT(I)%=T(I).T(N)%=TLL.
446 G%=L+1.H%=N-G.
447 'FOR'I%=1'STEP'1'UNTIL'H'DO'
448 T(I)%=TT(I+G).
449 G%=K-H.H%=L-G.
450 'FOR'I%=N-L'STEP'1'UNTIL'H'DO'
451 T(I)%=TT(I+G).
452 'FOR'I%=1'STEP'1'UNTIL'K'DO'
453 T('IF'C1'LE'C2'THEN'H+I'ELSE'N-I)%=TT(I).
454 'GOTO'A.
455 'END'C1.
456 'END' L.
457 'END' K.
458 ROTATION%
459 'FOR'I%=N'STEP'-1'UNTIL'2'DO'
460 T(I)%=T(I-1).
461 T(1)%=TN.
462 'END' J.
463 'IF'SF'THEN''BEGIN'
464 SF%='FALSE'.'GOTO'A'END'.
465 TN%=T(N).G%=0.
466 'FOR'I%=1'STEP'1'UNTIL'N'DO'
467 'BEGIN'
468 T1%=T(I).
469 G%=G+C(TN,T1).
470 S(TN,T1)%=S(T1,TN)%=1.
471 TN%=T1.
472 TU(IQ,I)%=T1.
473 'END'I.
474 GU(IQ)%=G.
475 'END'IQ.
476 'END' BODY LIN.
477 'END'RANDOM.
478 'END'LIN.
479
480                          'PROCEDURE'DOE(IETS,JA).
```

```
481                              'PROCEDURE'IETS.
482 'BOOLEAN'JA.
483 'BEGIN''COMMENT'DOET VOOR ELK SIGNAAL
484 DE PROCEDURE IETS(W,N),WAARIN W DE
485 HUIDIGE WYZER IN LYST1 OF LYST2
486 EN N HET AANTAL PENNEN WORDEN MEEGEGEVEN.
487 'INTEGER'W1,W2,NAAM,X,Y,N.
488 'BOOLEAN'P,Q,S,LAATSTE.
489 W1%=W2%=0.LAATSTE%='FALSE'.
490 LOOP1%    NAAM%=LYST1(W1).
491 'IF'JA'THEN''BEGIN'
492 LOOP3%     'IF'LYST1(W2)=NAAM'THEN''BEGIN'
493           W2%=W2+1.'IF'W2'LE'POINTER'THEN'
494           'GOTO'LOOP3'END'.
495           VECOSORT(LYST2,W1,W2-1).
496           DEKODEER(W1,X,Y,Q,S).
497           W2%=W1.
498 'END'JA.
499 LOOP2%    DEKODEER(W2,X,Y,P,S).
500           'IF'LYST1(W2)=NAAM'AND'
501           ((P'AND'Q)'OR'('NOT'P'AND''NOT'Q))
502           'THEN''BEGIN'W2%=W2+1.
503           'IF'W2'LE'POINTER'THEN''GOTO'LOOP2
504           'ELSE'LAATSTE%='TRUE''END'.
505           N%=W2-W1.
506           IETS(W1,N).
507           W1%=W2.
508           'IF''NOT'LAATSTE'THEN''GOTO'LOOP1.
509 'END'  DOE IETS.
510
511
512                              'PROCEDURE'OPTIMALISATIE(W,N).
513 'INTEGER'W,N.
514 'BEGIN''INTEGER'X,Y,KORTSTE,BEGINPUNT,I,J.
515 'INTEGER''ARRAY'C(1%N+1,1%N+1),
516 TOER(1%10,1%N+1),OPTOER(1%N),LENGTE(0%10).
517 'FOR'I%=1'STEP'1'UNTIL'N'DO''BEGIN'
518 C(I,N+1)%=C(N+1,I)%=10.C(I,I)%=E6.
519 'FOR'J%=1'STEP'1'UNTIL'I-1'DO'
520 C(I,J)%=C(J,I)%=AFSTAND(W+I-1,W+J-1)
521 'END' I.C(N+1,N+1)%=E6.
522 LIN(10,N+1,C,TOER,LENGTE).
523 KORTSTE%=1.
524 'FOR'I%=1'STEP'1'UNTIL'10'DO'
525 'IF'LENGTE(I)'LT'LENGTE(KORTSTE)'THEN'
526 KORTSTE%=I.
527 'FOR'I%=1'STEP'1'UNTIL'N+1'DO'
528 'IF'TOER(KORTSTE,I)=N+1'THEN'
529 BEGINPUNT%=I.
530 J%=1.
531 'IF'BEGINPUNT'LT'N+1'THEN'
532 'FOR'I%=BEGINPUNT+1'STEP'1'UNTIL'N+1'DO'
533 'BEGIN'OPTOER(J)%=LYST2(W+TOER(KORTSTE,I)-1).
534 J%=J+1'END'.
535 'IF'BEGINPUNT'GT'1'THEN'
536 'FOR'I%=1'STEP'1'UNTIL'BEGINPUNT-1'DO'
537 'BEGIN'OPTOER(J)%=LYST2(W+TOER(KORTSTE,I)-1).
538 J%=J+1'END'.
539 'FOR'I%=1'STEP'1'UNTIL'N'DO'
540 LYST2(W+I-1)%=OPTOER(I).
```

```
541 'END'OPTIMALISATIE.
542
543                              'PROCEDURE'STREEP.
544 'BEGIN''INTEGER'I.
545 NLCR.REGEL%=REGEL+1.
546 'FOR'I%=1'STEP'1'UNTIL'50'DO'PRSYM(65).
547 'END' STREEP.
548
549
550                              'PROCEDURE'UITVOER(W,N).
551 'INTEGER'W,N.
552 'BEGIN''INTEGER'X,LENGTE,Y,I,STAP.
553 'BOOLEAN'P,S.
554 'IF'REGEL-N'GT'45'THEN''BEGIN'
555 NEWPAGE.BLAD%=BLAD+1.REGEL%=3.
556 PRINTTEXT('('BEDRADINGSLIJST BAKALG IKO AMSTERDAM')').
557 PRINTTEXT('('          BLAD NO% ')').
558 ABSFIXT(4,0,BLAD).NLCR.
559 NLCR.
560 PRINTTEXT('('VAN PEN%        NAAR PEN%   ')').
561 PRINTTEXT('('        LENGTE (CM)%       SIGNAALNAAM%   ')').
562 NLCR.REGEL%=REGEL+1.
563 STREEP.
564 'END'.
565 STAP%=3.
566 I%=-2.
567 LOOP%
568 I%=I+STAP.
569 'IF'STAP=3'THEN'STAP%=-1'ELSE'STAP%=3.
570 'IF'I'GT'N+3'THEN''GOTO'AF.
571 'IF'I'LT'1'OR'I'GT'N-1'THEN''GOTO'LOOP.
572 NLCR.REGEL%=REGEL+1.
573 DEKODEER(W+I-1,X,Y,P,S).
574 PRINTKONPEN(X,Y,S).
575 TAB.
576 DEKODEER(W+I,X,Y,P,S).
577 PRINTKONPEN(X,Y,S).
578 SPACE(1).
579 TAB.
580 SPACE(6).
581 LENGTE%=ENTIER(AFSTAND(W+I,W+I-1)
582 *12/38+80)'DIV'40.
583 DRAADJE(LENGTE)%=DRAADJE(LENGTE)+1.
584 ABSFIXT(4,0,LENGTE*4).
585 TAB.
586 'IF'I=1'THEN''BEGIN'
587 TAB;PRINTNAAM(LYST1(W),P).
588 'END' I=1.
589 'GOTO'LOOP.
590 AF%
591 STREEP.
592 'END' UITVOER.
593
594
595 REGELNR%=1.FOUTJE%='FALSE'.
596 REGEL%=100.BLAD%=0.
597 SYM1%=SYM2%=SYM3%=SYM4%=SYM5%=93.
598 BEGINOFPROG%
599
600 BANDLEZEN.
```

```
  601
  602 EINDEBAND%
  603 'IF'FOUTJE'THEN''BEGIN'NLCR.
  604 PRINTTEXT('('GEEN VERDERE EXEKUTIE WEGENS FOUTEN')').
  605 'GOTO'EINDE  'END'.
  606 POINTER%=POINTER-1.
  607
  608 VEC2QSORT(LYST1,LYST2,0,POINTER).
  609 DOE(OPTIMALISATIE,'TRUE').
  610 'FOR'SYM1%=2'STEP'1'UNTIL'50'DO'
  611 DRAADJE(SYM1)%=0.
  612 DOE(UITVOER,'FALSE').
  613 NEWPAGE.
  614 PRINTTEXT('('VOOR DEZE BAK ZYN NODIG%')').
  615 NLCR.NLCR.
  616 'FOR'SYM1%=2'STEP'1'UNTIL'50'DO'
  617 'IF'DRAADJE(SYM1)'GT'0'THEN'
  618 'BEGIN'NLCR.
  619 ABSFIXT(5,0,DRAADJE(SYM1)).
  620 PRINTTEXT('('DRAADJES VAN ')').
  621 ABSFIXT(3,0,4*SYM1).
  622 PRINTTEXT('('CENTIMETER')').
  623 'END' SYM1.
  624 NLCR.NLCR.
  625 PRINTTEXT('('HET TOTALE AANTAL DRAADJES IS%')').
  626 SYM2%=0.
  627 'FOR'SYM1%=2'STEP'1'UNTIL'50'DO'
  628 SYM2%=SYM2+DRAADJE(SYM1).
  629 ABSFIXT(6,0,SYM2).
  630
  631
  632
  633
  634
  635 EINDE%
  636
  637
  638 NEWPAGE.
  639 PRINTTEXT('('DEZE BEDRADINGSLIJST IS GEMAAKT DOOR HET
  640 BEDRADINGSPROGRAMMA BAKALG,DAT DE DRAADLENGTE
  641 MINIMALISEERT MET HET R-OPT  ALGORITME VAN
  642 -LIN.
  643
  644 HET AANTAL BEDRADE PENNEN IN DEZE BAK IS%')').
  645 ABSFIXT(8,0,POINTER+1).NLCR.
  646 NLCR.
  647 PRINTTEXT('('DAT WAS HET DAN WEER
  648 VOOR DEZE KEER....
  649 HOEWEL....')').
  650
  651
  652 'END'
730302      17 UUR      47
    10 MILLI-UUR VERTAALTYD
730302      17 UUR      48
     0 MILLI-UUR REKENTYD
```