

**DIE ONTWIKKELING VAN 'N GEBRUIKERSVRIENDELIKE TRANSSEKOPINGS-
VIR IBM-AANPASBARE PERSOONLIKE REKENAARS.**

N D DU PREEZ

M.Ing.(Stellenbosch), B.Ing.(Stellenbosch)
Departement Bedryfsingenieurswese, Banhoekweg,
Universiteit van Stellenbosch, Stellenbosch 7600
Suid-Afrika

J C VAN DER MERWE

B.Ing.(Stellenbosch), N.D.I.(Stellenbosch)
Departement Bedryfsingenieurswese, Banhoekweg,
Universiteit van Stellenbosch, Stellenbosch 7600
Suid-Afrika

OPSOMMING

Die artikel beskryf die implementering van 'n gebruikersvriendelike transsekeringspakket, NET1, geskryf in Turbo Pascal vir IBM-aanpasbare persoonlike rekenaars. Die pakket is gebaseer op 'n reeds bestaande rekenaarprogram wat op die "out-of-kilter"-algoritme gebaseer is. Die validering van die program asook tydtoetse en moontlike toepassings word bespreek.

ABSTRACT

This article reports the implementation of a userfriendly transshipment computer program, NET1, written in Turbo Pascal for IBM-compatible personal computers. The program is based on an existing computer program based on the out-of-kilter algorithm. The validation of the program as well as time tests and applications are discussed.

1. Inleiding

Hierdie artikel beskryf die ontwikkeling van programmatuur vir IBM-aanpasbare persoonlike rekenaars vir die transskepingsvraagstuk. Die beperkte transskepingsmodel is die algemeenste vorm van die minimum koste vloeimodelle wat ook die beperkte en onbeperkte transportprobleme en die personeeltoekenningsprobleem insluit. Hierdie modelle is gewild omdat hulle 'n wye reeks toepassings soos bv. die vervoer van goedere, die ontwerp van kommunikasiesisteme en die toekenning van werknemers aan take akkuraat modelleer [2]. Verder word hierdie modelle ook soms gebruik as deel van grootskaalse optimeringsondersoeke.

Die beperkte transskepingsmodel en sy spesialisasies is minimum koste vloeimodelle. Die doel van die model is om die vloei van 'n kommoditeit deur 'n netwerk te vind vanaf die bronnodes na die sinknodes op so 'n manier dat die vervoerkoste daarvan geminimeer word terwyl die beperkings wat aan die boë gekoppel is, bevredig word. Die netwerk is 'n gerigte grafiek bestaande uit 'n versameling nodes, N , en 'n versameling boë, A . Die versameling boë word voorgestel deur die geordende getallepaar (i, j) . Kortliks word die getallepaar (i, j) aangedui as k . Vir elke boog is daar 'n verskepingskoste per eenheidsvloei, c_k , 'n minimum toelaatbare vloei, l_k , en 'n maksimum toelaatbare vloei, u_k , gespesifiseer. Drie node tipes word aangetref:

- 1) Die bronnodes wat net boë het wat daarvan af weg lei. Gewoonlik word nie-nul afvoer met hierdie nodes geassosieer.
- 2) Die sinknodes wat net boë het wat daarheen lei. Gewoonlik word 'n nie-nul vraag met hierdie nodes geassosieer.
- 3) Die transskepingsnodes wat boë het wat daarheen lei en ander wat daarvan af weg lei.

Wiskundig kan die probleem soos volg uitgedruk word:

$$\min. \sum_{k \in A} c_k x_k$$

$$\text{o.a. } \sum_{K \in A \text{ met stert } i} x_k - \sum_{K \in A \text{ met kop } i} x_k = b_i, \quad i \in N,$$

$$l_k < x_k < u_k, \quad k \in A$$

waar $b_i =$ (aanbod as i 'n bronnode is; vraag as i 'n sinknode is; 0 andersins). Die notasie kom uit Bradley et.al. [3].

Sedert Hitchcock en Koopmans se baanbrekerswerk op die gebied van netwerkmodelle in die vroeë veertigerjare, het ontwikkeling op hierdie gebied vinnig gevorder. Wat die transskeepingsprobleem betref, bestaan daar vandag ses basiese benaderings om dit mee op te los nl. die primale simpleksmetode [3,6,8], die "out-of-kilter"-metode [1,2,9], die primaal-duale metode [2,3], die duale metode [2,3], die padmetode [2,3], die negatiewe siklusmetode [2,3] en die skaleringsmetode [2,3].

Weens die gewildheid van netwerkmodelle in Operasionele Navorsing en ook weens die geweldige opbloei wat persoonlike rekenaars in Suid-Afrika beleef, gekoppel aan die feit dat die persoonlike rekenaar rekenartegnologie finansieel binne bereik van die meeste firmas en instansies in Suid-Afrika gebring het, het dit nodig geword om netwerkprogrammatuur te ontwikkel vir 'n persoonlike rekenaar. Sodoende word netwerkmodelle dan tot die beskikking van alle gebruikers van persoonlike rekenaars gestel.

2. Die program

Die program is in Turbo Pascal (Weergawe 3.01) geskryf en kan op enige IBM-aanpasbare persoonlike rekenaar of ander persoonlike rekenaar wat Turbo Pascal gebruik, loop. Die program is gebaseer op die "out-of-kilter" algoritme en kan tot 10500 boë hanteer. Die algoritme is 'n aanpassing van 'n reeds bestaande FORTRAN "out-of-kilter"-program [10].

'n Oogmerk van die projek was om die rekenaarprogram so gebruikersvriendelik as moontlik te maak. Die program, NET1, is die produk van 'n langdurige ontwikkelingsproses en beskik tans oor die volgende eienskappe: 'n hoofkeusekaart waarvandaan die belangrikste hoof funksies bereik kan word, 'n redigeerder waarmee data gelaai of verander kan word, funksies vir die laai en ontlaai van datalêers na of vanaf 'n slapskyf, funksies vir die druk van datalêers en resultate met 'n drukker, funksies vir die vertoning van datalêers en resultate op die rekenaarskerm, 'n funksie wat die geheue skoonmaak en 'n funksie vir die beëindiging van die sessie.

Omdat daar in die praktyk meestal 'n groot hoeveelheid data ingelees moet word, is daar veral gepoog om die redigeerder, waarmee die data ingelees of verander word, so gebruikersvriendelik as moontlik te maak. Die redigeerder is so ontwerp dat die gebruiker 'n rowwe skets van die netwerk wat hy wil oplos, behoort te hê in plaas van 'n stel vergelykings soos wat bv. by Lineêre Programmeringsprogrammatuur vereis sou word. Dit vergemaklik die gebruiker se taak in die sin dat die probleem visueel voorgestel word in plaas daarvan dat 'n stel vergelykings en veranderlikes gedefinieer word. Die visuele voorstelling van 'n netwerkprobleem is dikwels makliker as wat dit sou wees as dit deur vergelykings gedefinieer moes word. Inligting wat verlang word, is die aantal boë en aantal nodes sowel as die kopnode, stertnode, maksimum kapasiteit of vloeitempo, minimum kapasiteit of vloeitempo en eenheidsvloekoste vir elke boog.

'n Eienskap van die redigeerder wat nuttig is om groot hoeveelhede data vinnig te lees, is die DATA FILL-funksie. Dit is 'n funksie wat soortgelyk is aan LOTUS-1-2-3 se DATA FILL-funksie. Die funksie word gebruik om groot hoeveelhede data wat op mekaar volg en met dieselfde inkrement van mekaar verskil, outomaties te genereer. Daar is ook gevind dat dit moontlik is om maklik programme te skryf wat datalêers skep wat deur NET1 gelees kan word. So is daar bv. 'n program, READ1, geskryf wat netwerkprobleme wat d.m.v. 'n ander program geskep is in 'n datalêer omskep wat deur NET1 gebruik kan word. 'n Ideale uitbreiding van die redigeerder sou wees om die program so te

verander dat dit ten volle interaktief met die LOTUS 1-2-3 pakket is.

3. Sirkulasienetwerke

Die "out-of-kilter"-algoritme vereis dat die netwerk as 'n sirkulasienetwerk voorgestel word. 'n Sirkulasienetwerk word gekenmerk deur die feit dat dit 'n enkele bronnode en 'n enkele sinknode het wat benewens hulle verbinding met ander nodes in die netwerk ook direk aan mekaar verbind is. Kommoditeite vloei deur die netwerk van die bronnode na die sinknode en vanaf die sinknode deur die boog wat die bron- en sinknodes direk met mekaar verbind terug na die bronnode. Die grootte van die vloei deur die direkte verbindingsboog van die sink- na die bronnode toon dus die totale vloei wat deur die res van die netwerk gaan.

Die oorgrote meerderheid transskepingsprobleme en spesialisasies daarvan is egter nie natuurlike sirkulasienetwerke nie. Hierdie probleme moet dus aangepas word na sirkulasienetwerke voordat dit deur NET1 gebruik kan word. As 'n voorbeeld word gewys hoe 'n transportnetwerk bestaande uit n bronne en m sinke tot 'n sirkulasienetwerk omgeskep word. In figuur 1 word die oorspronklike transportprobleem getoon.

Hierdie probleem bestaan uit $n*m$ boë en $n+m$ nodes waar n die aantal bronne en m die aantal sinke is. Die sirkulasienetwerk word in figuur 2 getoon.

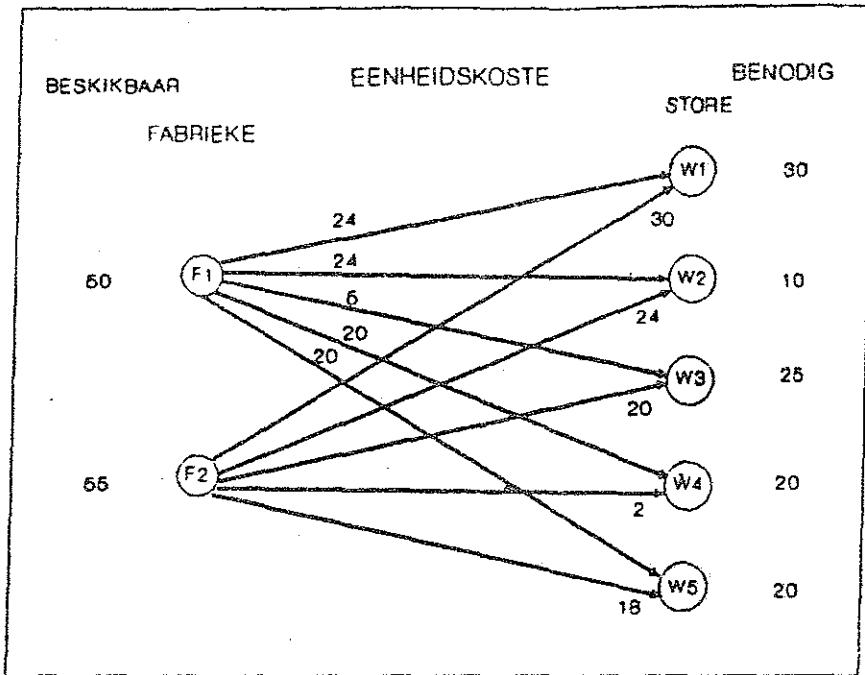


Fig. 1. Die transportprobleem.

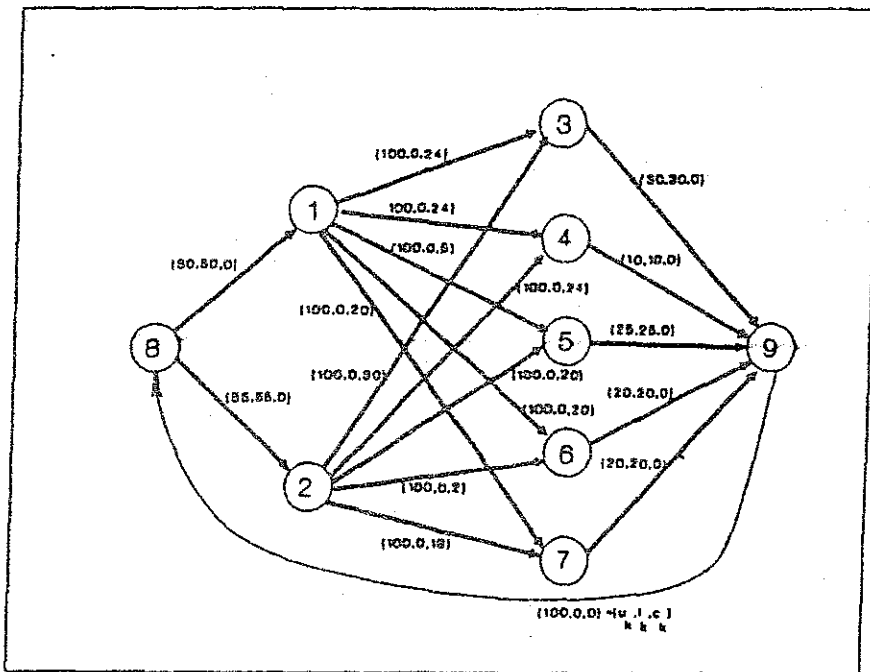


Fig. 2. Die aangevulde transportprobleem.

'n Superbronnnode (node 8) en supersinknode (node 9) sowel as die verbindingsboë tussen die superbronnnode en supersinknode en ander nodes, en die direkte verbindingsboog tussen die superbronnnode en super-sinknode is bygevoeg. Die sirkulasienetwerk bestaan dan uit

$n*m+n+m+1$ boë en $n+m+2$ nodes. Die voorbeeld kom uit Daellenbach et.al. [4].

4. Validasie

Om die program te toets asook 'n idee te kry van hoe lank die program loop, is vier tipes teksboekprobleme uit Daellenbach et.al. [4] en 'n aantal regte wêreld probleme bestaande uit tot 10 250 boë, getoets. Die teksboekprobleme wat getoets is, is 'n ten volle digte transportprobleem, 'n transskepingsprobleem, 'n toekenningsprobleem en 'n maksimum vloeioprobleem. Hierdie lopies is gedoen met 'n IBM aanpasbare persoonlike rekenaar met 'n 8088 mikroverwerker met 8 mHz verwerkingspoed.

4.1 Die Transportprobleem

Soos reeds verduidelik, is dit nodig om te sorg dat alle probleme tot sirkulasienetwerke omskep word. Die transportprobleem wat in figuur 1 en figuur 2 getoon is, is reeds bespreek. Die toekenningsprobleem word in tabel 1 uiteengesit.

Tabel 1. Toekenningsprobleem

Jaarlikse tranportkoste (R1000'e)

<u>Toegeken aan seksie</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
Voertuig 1	6	4	1	5
Voertuig 2	10	6	3	8
Voertuig 3	7	6	4	5
(Geen) 4	9	10	3	8

Die probleem kom daarop neer dat drie voertuie beskikbaar is. Vier seksies het egter elkeen 'n voertuig nodig. Tabel 1 toon die koste verbonde aan die toekenning van 'n voertuig aan 'n spesifieke seksie. Die probleem is om die voertuie so toe te ken dat die totale koste 'n minimum is. Die volledige sirkulasienetwerk van die probleem word in figuur 3 getoon.

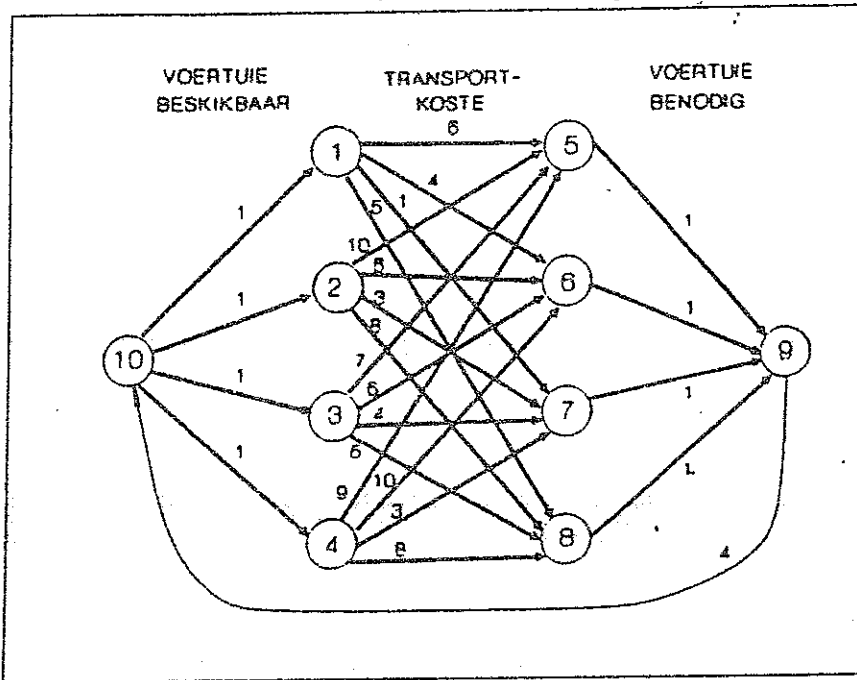


Fig. 3. Die aangevulde toekenningsprobleem.

4.2 Die Transskepingsprobleem

Die transskepingsprobleem word in figuur 4 getoon.

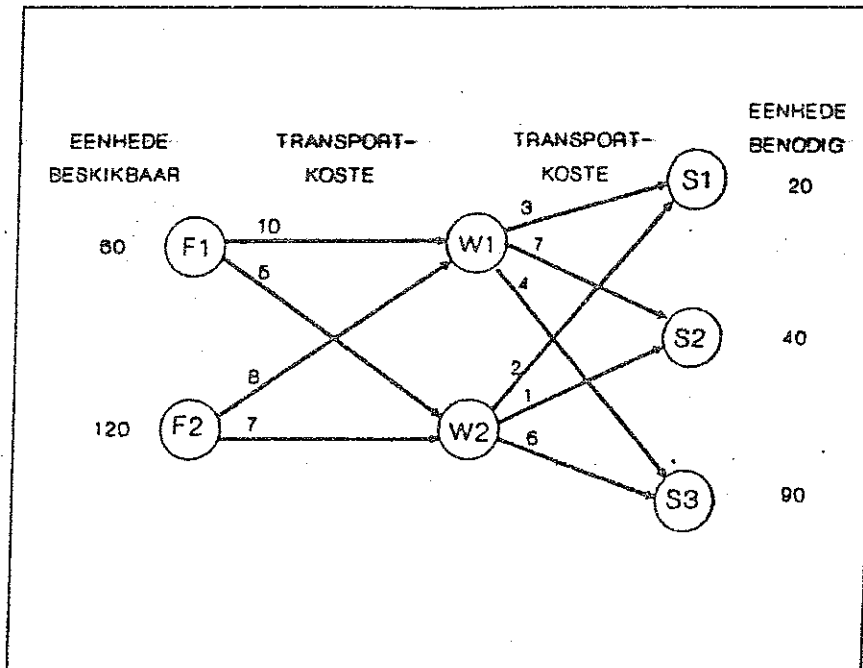


Fig. 4. Die transskepingsprobleem

Volgens die probleem is die maksimum kapasiteit van die transskepingsnode, W1, net 50 eenhede. Die transskepingsnode W2 gebruik verder self 30 eenhede van die kommoditeit wat by nodes F1 en F2 beskikbaar is en by nodes S1, S2 en S3 benodig word in hoeveelhede soos aangetoon in figuur 4. Die volledige sirkulasienetwerkvoorstelling van die probleem word in figuur 5 getoon.

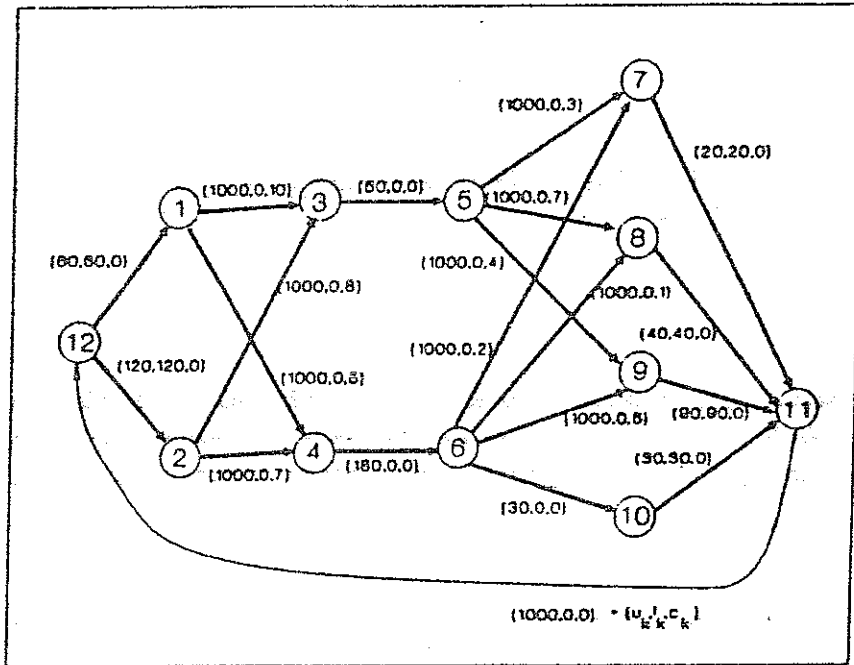


Fig.5. Die aangevulde transskepingsprobleem.

Nodes 5 en 6 is bygevoeg om vloeibeperkings by W1 en W2 in figuur 4 voor te stel. Node 10 in figuur 5 stel die verbruik van 30 eenhede van die kommoditeit by node W2 in figuur 4 voor.

4.3 Die maksimum vloeiprobleem

Die maksimum vloeiprobleem word in figuur 6 getoon. Die getalle by elke boog se pyltjie stel die maksimum aantal kommoditeite voor wat in daardie boog mag vloei. Die volledige sirkulasienetwerk word in figuur 7 getoon.

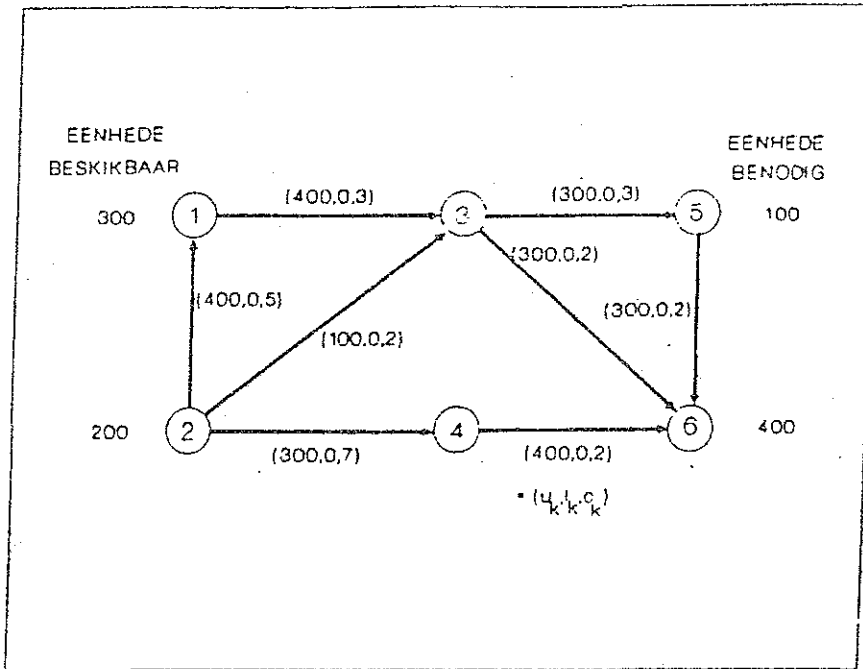


Fig. 6. Die maksimum vloei probleem.

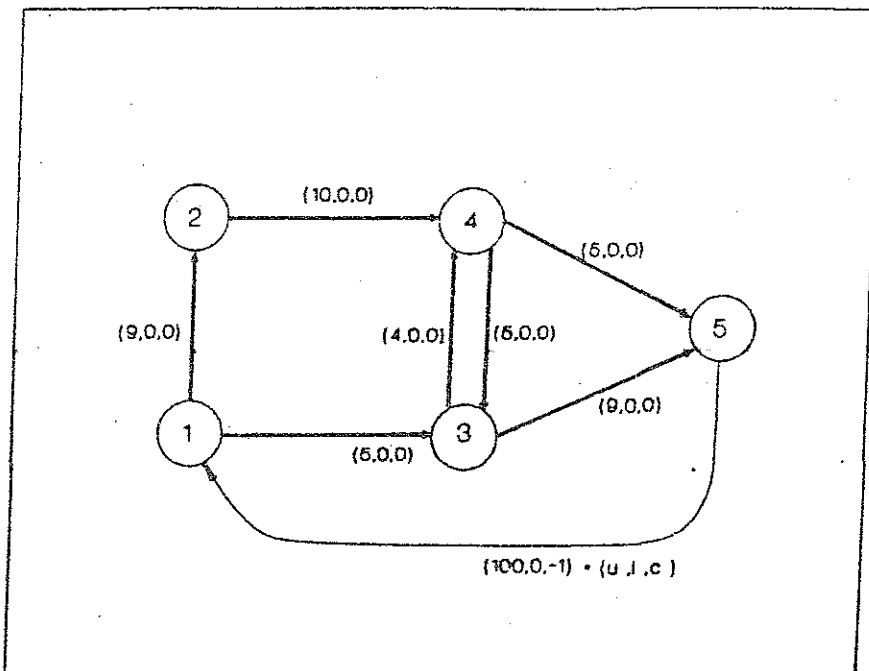


Fig. 7. Die aangevulde maksimum vloei probleem.

Die maksimum aantal kommoditeite in die boog word deur u_k voorgestel, die minimum aantal kommoditeite in die boog word deur l_k voorgestel en die eenheidskoste vir die vervoer van kommoditeite in die spesifieke boog word deur c_k voorgestel.

4.4 Regte Wêreld Probleme

Sewe regte wêreld probleme is getoets. Die probleme is almal ten volle digte onbeperkte (E. uncapacitated) transportprobleme en het van 33 to 10 250 boë. Die looptye en antwoorde is getoets teen die tye en antwoorde van bestaande transportprogrammatuur wat op die UNIVAC-1100 rekenaar van die Universiteit van Stellenbosch beskikbaar is. Die pakket was oorspronklik 'n IBM-program wat in 1975 aangepas is vir die UNIVAC 1100. Dit gebruik die MODI-metode met ry minimum koste. Die probleme is geskep d.m.v. 'n reeks programme wat deur Dixon et.al. [5] ontwikkel is om die vervoerpatroon van mielies in Suid-Afrika te ondersoek.

Uit 'n vergelyking tussen die resultate van die oorspronklike algoritme wat deur Dixon gebruik is en die resultate van die NET1-program, blyk dit dat die maksimum koste en die totale hoeveelheid vloei ooreenstem. Daar is egter verskille in die aantal kommoditeite wat aan sommige boë toegeken word. Hieruit kan afgelei word dat daar meer as een optimum vloei konfigurasie is aangesien die totale koste verbonde aan beide konfigurasies dieselfde is.

Uit figuur 8 kan gesien word dat die looptye van die transportprogram van die UNIVAC 1100 ongeveer eksponensieel styg soos die aantal boë toeneem. Verder is die looptye van NET1 van 'n faktor 3 vir die 10250-boog probleem tot 'n faktor 28 vir die 200-boog probleem groter as die looptye van die transportprogram van die UNIVAC 1100. Vir die 33-boog probleem is die NET1 met die CW-16 'n uitsonderlike 107 keer stadiger as die transportprogram van die UNIVAC 1100. Die verskil in die faktor kan toegeskryf word aan die feit dat die algoritmes wat gebruik word om die probleem mee op te los, vanmekaar verskil. Die verskil in die aantal iterasies wat elke algoritme vir 'n probleem uitvoer, sal dus grootliks wissel van probleem tot probleem. Daar is ook 'n groot afname in die faktor namate die aantal boë van die probleem toeneem. Die feit dat die transportprogram van die UNIVAC 1100 net 3 keer vinniger as NET1 met die CW-16 is vir die 10250-boog probleem, toon dat die transportpakket baie oneffektief is vir netwerkprobleme in vergelyking met 'n gespesialiseerde netwerk algoritme soos die "out-of-

kilter"-algoritme.

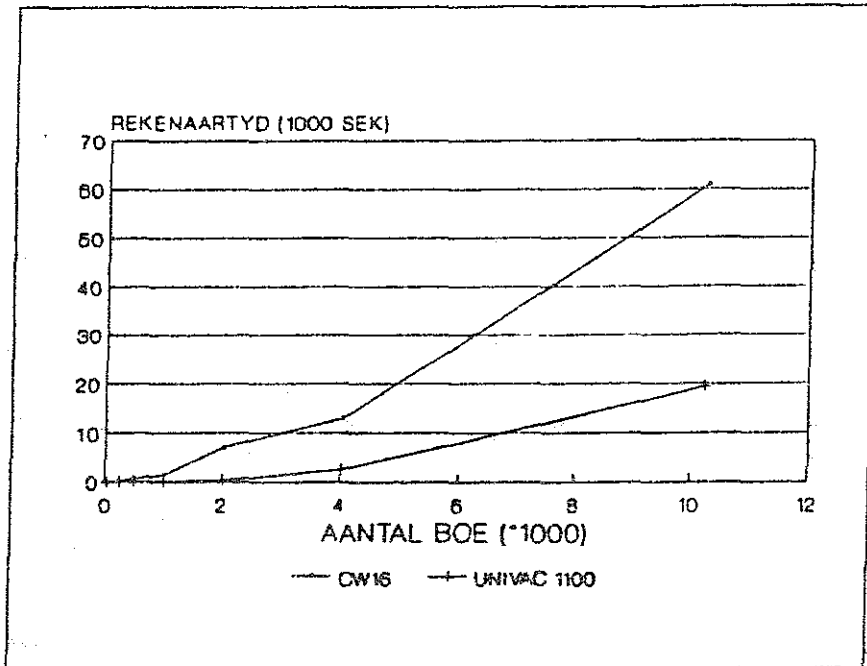


Fig. 8. Berekeningstye teenoor aantal boë.

Hierdie tydwaardes gee net naastenby 'n gevoel van hoe NET1 met 'n transportprogram van die UNIVAC 1100 vergelyk. 'n Idee van hoe NET1 met 'n CW-16 persoonlike rekenaar en moderne primale simpleks gebasseerde programmatuur en die CDC 6600 rekenaar met mekaar vergelyk, kan gekry word deur op te merk dat die transportprogram en die UNIVAC 1100 tot 90 keer stadiger as die GNET-program [3] met die CDC 6600 rekenaar is. Dus kan NET1 tot 270 keer stadiger wees as moderne programmatuur met 'n moderne hoofraamrekenaar vir 'n 10250-boog probleem.

5. Toepassings

Die feit dat die berekeningstyd van NET1 met die CW-16 so lank is, het nie 'n groot impak op die bruikbaarheid van die program nie. Die berekeningstyd raak net krities waar daar vir rekenaartyd betaal word soos by die meeste hoofraamrekenaars die geval is. Die enigste noemenswaardige beperkingsfaktor van NET1 is die beperkte hoeveelheid geheue wat 'n persoonlike rekenaar bied.

Dixon et.al. [5] het egter aangetoon dat vir 'n toepassing waarin die ten volle digte transportprobleem gebruik word, dit moontlik is om die probleem te vereenvoudig van 'n vlak van 38218 moontlike boë tot slegs 3995 moontlike boë sonder om akkuraatheid noemenswaardig in te boet. Dit word gedoen deur die gebied wat bestaan uit produksie- en verbruikerspunte in ruite te verdeel en dan die massamiddelpunt van so 'n ruit as 'n produksie- of verbruikerspunt te neem. Sodoende word 'n growwer benadering met minder moontlike boë verkry, wat nogtans 'n baie akkurate antwoord gee. Die meeste toepassings van hierdie aard is in elk geval op inligting gegrond wat net by benadering korrek is. So 'n geaggregeerde benadering is dus meestal geregverdig. Dus is dit moontlik om selfs probleme wat aanvanklik te groot is om d.m.v. NET1 met 'n persoonlike rekenaar op te los deur eksperimentering te vereenvoudig tot 'n probleem wat wel met voldoende akkuraatheid deur sodanige programmatuur opgelos kan word.

Die "out-of-kilter"-algoritme waarop NET1 geskoei is, is 'n baie aanpasbare algoritme. Nie net kan die transskepingsfamilie probleme hanteer word nie, maar kortste pad en maksimum vloei probleme met invoerdata wat reg aangebied word, kan ook behandel word. Die vorm waarin 'n maksimum vloei probleem se data aangebied behoort te word, is reeds deur een van die teksboekprobleme geïllustreer in figuur 6. Die vorm waarin 'n kortste padprobleem se data aangebied behoort te word, word in figuur 9 getoon. (bi) dui die aantal eenhede beskikbaar by node i aan. Die doel hier is om die kortste pad tussen node 1 en node 3 te vind. Om die probleem as 'n sirkulasienetwerk voor te stel, is dit egter nog nodig om 'n verbindingsboog vanaf node 3 na node 1 te trek met $c_k = 0$ en $u_k = 1$.

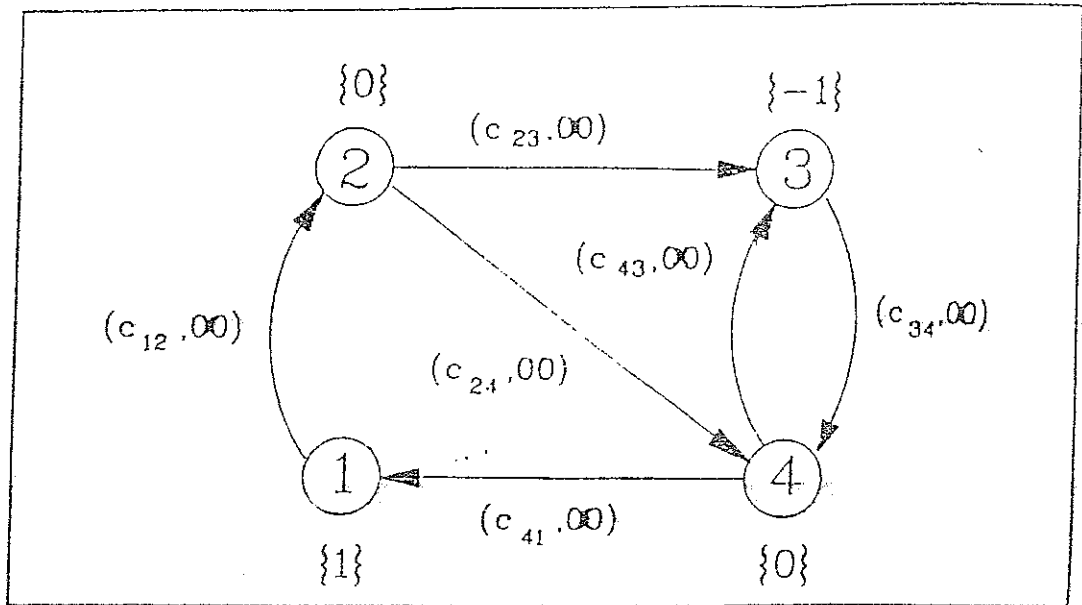


Fig. 9. Die kortste padprobleem.

6. Opsommend

Die groot voordeel van 'n program soos NET1 wat met 'n persoonlike rekenaar loop, is sy koste-effektiwiteit. M.b.v. programmatuur soos hierdie word fasiliteite wat voorheen net deur groot instansies of by geleentheid deur kleiner instansies gebruik kon word, binne bereik geplaas van almal wat 'n persoonlike rekenaar van R3000 kan bekostig teen 1987 pryse.

Omdat netwerkalgoritmes so wyd toepasbaar en maklik verstaanbaar is vir selfs diegene wat nie 'n gevorderde wiskundige agtergrond het nie, behoort programmatuur soos NET1 met 'n bietjie oefening ook handig deur persone in bestuursposisies gebruik te kan word. 'n Boek wat nuttige, vinnige en kragtige wenke in die verband gee is Woolsey en Swanson se Quick and Dirty manual [10].

VERWYSINGS

- [1] R.S. Barr, F. Glover en D. Klingman, "An Improved Version of the Out-of-Kilter Method and a Comparative Study of Computer Codes", Mathematical Programming, Vol. 7, (1974), pp. 60-86.
- [2] G.H. Bradley, "Survey of Deterministic Networks", AIIE Transactions, Vol. 7, (1975), pp. 222-234.
- [3] G.H. Bradley, G.G. Brown en G.W. Graves, "Design and Implementation of Large Scale Primal Transshipment Algorithms", Management Science, Vol. 24, (1977), pp. 1-34.
- [4] H.G. Daellenbach, et.al., Introduction to Operations Research Techniques, Allyn and Bacon, Boston, 1983.
- [5] E. Dixon en N.D. du Preez, "Optimering van die Vervoerpatroon van Mielies in Suid-Afrika", Verslag van die IBI, Universiteit van Stellenbosch, Suid-Afrika.
- [6] F. Glover, D. Karney en K. Klingman, "Implementation and Computational Comparisons of Primal-Dual Computer Codes for Minimum Cost Network Flow Problems", Networks, Vol. 4, (1974), pp. 191-212.
- [7] F. Glover, D. Karney, D. Klingman and A. Napier, "A Computational Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems", Management Science, Vol. 20, (1974), pp. 793-819.
- [8] M.D. Grigoriades, "An Efficient Implementation of the Network Simplex Method", Mathematical Programming, Vol. 26, (1986), pp. 83-111.
- [9] G.H. Kennington en R.V. Helgason, Algorithms for Network Programming, Wiley, New York, 1980.
- [10] R.E.D. Woolsey en H.S. Swanson, Operations Research for Immediate Application: a Quick and Dirty Manual, Harper and Row, New York, (1975).