Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

MC SYLLABUS 48.2

COLLOQUIUM COMPLEXITEIT EN ALGORITMEN

DEEL 2

P.M.B. VITÁNYI (red.)
J. van LEEUWEN (red.)
P. van EMDE BOAS (red.)

MATHEMATISCH CENTRUM AMSTERDAM 1982

1980 Mathematics subject classification: 68C05, 68C25, 68C40, 68H05

ACM-Computing Reviews-category: 5.25, 5.26, 4.34

ISBN 90 6196 246 3

INHOUD

COMPLEXITEIT EN ALGORITMEN, DEEL 2.

ADRESSEN VAN DE AUTEURS

P. van Emde Boas	Universiteit van Amsterdam, Subfaculteit Wiskunde, Roetersstraat 15, 1018 WB Amsterdam.
B.J. Lageweg	Mathematisch Centrum, Kruislaan 413, 1098 SJ Amsterdam.
C.G. van der Laan	Rijksuniversiteit Groningen, Rekencentrum, Universiteitscomplex Paddepoel, Postbus 800, 9700 AV Groningen.
E.L. Lawler	University of California Berkeley, Computer science division, Berkeley, CA 94720, U.S.A.
H.W. Lenstra, Jr.	Universiteit van Amsterdam, Subfaculteit Wiskunde, Roetersstraat 15, 1018 WB Amsterdam.
J.K. Lenstra	Mathematisch Centrum, Kruislaan 413, 1098 SJ Amsterdam.
A. Schrijver	Universiteit van Amsterdam, Instituut voor Actuariaat en Econometrie, Jodenbreestraat 23, 1011 NH Amsterdam.
A.H.G. Rinnooy Kan	Erasmus Universiteit Rotterdam, Econometrisch Instituut, Burgemeester Oudlaan 50, 3062 PA Rotterdam.

VOORWOORD

In het academisch jaar 1980/1981 werden op het Mathematisch Centrum een dertiental voordrachten gehouden over *Complexiteit en Algoritmen*. Bij de voordrachten werden syllabi uitgerekt. Deze bundel bevat de door de auteurs bijgewerkte en gecorrigeerde versies daarvan. De bijdragen zijn naar onderwerp gebundeld in vijf groepen verdeeld over twee delen, waarvan dit het tweede is. In plaats van op de verschillende onderwerpen in te gaan laten wij de inhoudsopgave voor zichzelf spreken en wijden dit voorwoord aan een korte inleiding tot het onderhavige vakgebied.

Wie rekent zal gewoonlijk een bepaald algoritme volgen. Door preciese analyse kan men soms een uitspraak doen over het aantal handelingen (bewerkingen, assembler instructies) dat nodig is om een probleem van formaat "n" volgens het gekozen algoritme tot een oplossing te brengen. Tesamen met overoverwegingen inzake de vereiste I/O transporten, verkrijgt men zo een redelijke indruk van de te verwachten rekentijd (en/of geheugengebruik) van een algoritme op een realistisch machinemodel. Zo'n analyse is vaak nodig om tot een duidelijk onderscheid te komen tussen efficiënte en niet zo efficiënte algoritmen voor eenzelfde probleem.

Sinds de vijftiger jaren is men zich er meer en meer van bewust geworden dat een nauwkeurige evaluatie van de efficiëntie van gebruikte rekenmethodes doorslaggevend kan zijn voor het welslagen van te maken software. Een steeds terugkerende ervaring is echter dat er, hoe slim men ook programmeert, soms geen noemenswaardige vooruitgang in efficiëntie te behalen lijkt in de algoritmische oplossing van een probleem. We schijnen dan op een intrinsieke *complexiteit* van het probleem te stuiten, die ieder denkbaar algoritme dwingt tot het uitvoeren van een zeker minimum aantal bewerkingen. Deze indruk kan onjuist zijn, maar alleen door preciese analyse kan men erachter komen of er geen betere oplossingen bestaan. De voortdurende pogingen om steeds maar weer algoritmen te vinden die hun doel met mindere bewerkingen dan tot nog toe bekend bereiken, zijn een methode om de complexiteit van problemen te bepalen en om uiteindelijk tot praktisch efficiënte rekenmethodes te komen. Wat vroeger misschien een slimme programmeertruc leek, blijkt dan niet zelden een algemene techniek te zijn om goede algoritmen toch weer te versnellen. Zo kent men thans tal van technieken (zoals depth-first search, path compression, Fast Fourier Transform, dynamization) die, hoewel alleen door de theorie op hun juiste waarde getaxeerd, in praktisch

programmeerwerk hun diensten kunnen bewijzen. En nieuwe berekeningsmodellen (zoals de hardware realisatie van algoritmen op chips) geven steeds weer nieuwe uitganspunten voor een onderzoek van de wezenlijk haalbare efficiëntie van algoritmen.

In het colloquium is een poging gedaan om de vele facetten van *complexiteit en algoritmen* wat grotere bekendheid te geven. In een aantal voordrachten werden door de sprekers vooral recente vorderingen in dit gebied geëxposeerd, al dan niet met kritische kanttekeningen over hun betekenis zoals die thans gezien wordt. De bedoeling is dat dit zal bijdragen tot een duidelijk beeld van de mogelijkheden en beperkingen van algoritmisch analyse voor de ontwerper van rekenmethoden.

D. ALGEBRAÏSCHE EN ANALYTISCHE COMPLEXITEIT

BEREKENINGSCOMPLEXITEIT VAN BILINEAIRE EN KWADRATISCHE VORMEN

P. VAN EMDE BOAS

1. INLEIDING

In 1969, toen Complexiteitstheorie nog een ongeboren vak was, verraste [167] de Duitse wiskundige Volker Strassen de wereld met een stel formules die aantonen dat het mogelijk is met 7 vermenigvuldigingen het product van twee matrices van afmetingen 2 bij 2 te vormen. Omdat bovendien geen gebruik wordt gemaakt van de commutativiteit van vermenigvuldiging zijn deze formules ook toepasbaar in het geval dat de elementen van de matrix afkomstig zijn uit een niet commutatieve ring (zoals een ring van matrices bijvoorbeeld). Op grond hiervan constateert men dat matrixvermenigvuldiging van matrices van 2^k bij 2^k uitgevoerd kan worden met niet meer dan 7^k vermenigvuldigingen, hetgeen uiteindelijk leidt tot een bovengrens voor de arithmetische complexiteit van dit probleem van de orde $n^{2.81}$, waarbij de exponent 2.81 verkregen is als de waarde van $^2\log(7)$. Dit getal trad al spoedig op in vele andere complexiteitsgrenzen, zoals matrixinversie (STRASSEN [167]), diverse andere matrixbewerkingen (BUNCH & HOPCROFT [30], SCHÖNHAGE [148]), bewerkingen op Boolse matrices zoals transitieve afsluiting (zie bijv. [112]), terwijl er zelfs een "bedrog" reductie bestaat om matrices in de min, + - algebra te vermenigvuldigen via Strassen's identiteiten (zie YUVAL [201] of ROMANI [139]); hiermee komen de problemen als kortste afstanden in een graaf onder het bereik van deze methoden. De exponent treedt ook op in een algoritme voor het herkennen van contextvrije talen (VALIANT [181]).

Strassen's resultaat, dat aantoont dat de standaardmethode voor het vermenigvuldigen van matrices niet optimaal is, vormt het motiverende voorbeeld in de sinds 1969 gegroeide theorie van de complexiteit van bilineaire vormen, die op zijn beurt een onderdeel is van het veel uitgebreidere terrein van de Algebraïsche Complexiteitstheorie. Deze theorie blijkt een ideaal proefveld waar wiskundigen (in het bijzonder algebraïci) hun begrippenapparaat vanuit hoger standpunt kunnen aanwenden tot het verkrijgen van

interessante inzichten, terwijl anderzijds de minder wiskundig getrainde informatici met veel rekenwerk en moeizame toepassingen van elementaire lineaire algebra tot resultaten weten te komen.

Tot 1978 leidde het verkregen inzicht in de aard van het probleem niet tot een wezenlijke verbetering van Strassen's resultaat, en het getal 2.81 verwierf aldus een stevige positie in de verzameling van wereldconstanten. In 1978 slaagde V. Pan er echter in te komen met een nieuw schema dat aanleiding gaf tot een lagere exponent: (2.79) [116], en sindsdien is op dit terrein de rust niet weergekeerd. Gezamenlijke arbeid van een groep Italianen [13,15], SCHÖNHAGE [152], PAN & WINOGRAD [80,117,120] heeft er toe geleid dat de waarde van de matrixvermenigvuldigingsexponent thans† gelegen is bij 2.52; dit valt op te maken uit een inscriptie in het gastenboek van het Mathematisches Forschungsinstitut Oberwolfach, deel 4, luidende: "As of 21.24 hr. of October 26, 1979 the best known exponent for matrix multiplication is 2.521812716" VICTOR PAN & SHMUEL WINOGRAD. Geruchten, als zou de grens inmiddels alweer verbeterd zijn tot 2.51 of zelfs 2.49 schijnen te berusten op een inmiddels als foutief achterhaald bewijs [199]†.

In deze voordracht wil ik een poging doen om, bij wijze van illustratie van dit deel van de complexiteitstheorie, uit te leggen wat er aan wiskunde schuil gaat achter deze jacht op de matrixvermenigvuldigingsexponent. Ik zal U moeten uitleggen wat dit getal precies betekent (waarna U zich geen zorgen behoeft te maken dat U Uw duur gekochte programmatheek voor het oplossen van lineaire stelsels naar de prullebak dient te verwijzen). Ik zal U het begrip *tensorrang*, dat in deze theorie een belangrijke rol blijkt te spelen, niet kunnen onthouden, al was het maar om de vermaarde symmetriestelling voor matrixvermenigvuldiging te kunnen verklaren. Tenslotte wil ik U laten zien hoe de verbeterde grenzen ontstaan uit een samenvloeiing van drie onderling onafhankelijke nieuwe ideeën.

2. BEREKENING EN PROGRAMMA

Beschouw de welbekende wortelformule voor de oplossingen van de vierkantsvergelijking $ax^2 + bx + c = 0$: $x_{1,2} = (-b \pm \sqrt{b^2 - 4ac})/2a$.

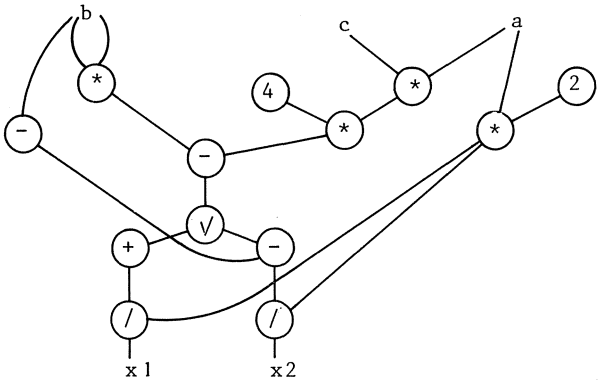
† Aangezien de vormgeving van deze voordracht sterk gericht is op de toestand ten tijde van de bijeenkomst te Oberwolfach in Oct. 1979, is er van afgezien bij de revisie de nieuwste beschikbare informatie in de tekst te verwerken. Inmiddels achterhaalde uitspraken zijn met een † -teken aangeduid; voor een korte uiteenzetting over de nieuwste ontwikkelingen, verwijs ik naar paragraaf 12: Ontwikkelingen sinds 1980.

Deze formule zondigt tegen practisch alles wat wij in programma's plegen tegen te komen aan syntactische eisen; men verwacht eerder een rijtje opdrachten in de geest van

```
h1 = b * b
h2 = a * c
h3 = 4 * h2
h4 = h1 - h4
h5 = sqrt h4
h6 = -b
h7 = h6 + h5
h8 = h6 - h5
h9 = 2 * a
x1 = h7/h9
x2 = h8/h9
```

Iedere regel in dit programma laat zien hoe een nieuwe waarde berekend wordt door het toepassen van een operator op een of twee operanden die, hetzij constanten (2,4) hetzij gegevens (a, b of c), hetzij eerder berekende resultaten zijn (h1,...,h9), terwijl de te berekenen grootheden (x1,x2) tegen het einde van het programma optreden.

De berekening had zich ook laten formaliseren in de vorm van een gerichte acyclische graaf, voorzien van de nodige labels als hieronder aangegeven:



In de ordening in de graaf is duidelijk welke waarden voor welke andere grootheden dienen te worden berekend, terwijl het programma als het ware een expliciete ordening aan de vorming der resultaten oplegt, ook waar deze volstrekt irrelevant is (zoals bij de vorming van h1 en h2). Anderzijds laat het formalisme van het programma toe aan te geven dat het resultaat $b^2 - 4ac$ twee keer berekent dient te worden; in de graaf zou men in feite dan de knoop dienen te herhalen.

Bij het formaliseren is het van belang in te zien dat er een verschil bestaat tussen het programma enerzijds, dat, weergegeven in de vorm van een reeks opdrachten dan wel een graaf, aangeeft welke operaties dienen te worden uitgevoerd op welke gegevens, constanten of eerdere resultaten, en anderzijds de rij waarden die bij het eventuele uitvoeren van het programma op elementen uit een gegeven algebraïsche structuur zullen worden gevormd. V. Strassen geeft in zijn formaliserende artikel 'Berechnung und Programm I' [169] aan deze reeks waarden de naam *berekening*.

In deze formalisatie gaan we uit van een algebraïsche structuur, waarop een stel operaties gedefinieerd zijn; hierbij is de naam van de operator en het aantal operanden essentiëel. Constanten worden beschouwd als additionele operatoren in de algebra met 0 operanden. Een programma is nu een rij regels, waarbij iedere regel bestaat uit een operator, gevolgd door het bijbehorende aantal operanden; de operanden zijn hetzij gegevens, hetzij regelnummers van eerdere regels. Een berekening van dit programma ontstaat door het programma te interpreteren over een concrete algebra van het goede type (dwz. de bijbehorende operatoren hebben in deze algebra het juiste aantal argumenten, inclusief de constanten die gewoon in de concrete algebra een betekenis dienen te hebben), en door voor de gegevens bijbehorende elementen in de algebra te kiezen.

Opgemerkt dient te worden dat Strassen zich bij zijn formalisatie beperkt tot éénsoortige algebra's (de drager van de algebra kent slechts één type). Het ligt meer voor de hand om te kijken naar meersoortige algebra's waarbij er sprake is van meerdere verzamelingen met operaties ertussen (denk bijv. aan een vectorruimte over een lichaam, waarbij de algebraïsche structuur bestaat uit een lichaam en de vectorruimte, en waarbij operaties als + en - gedefiniëerd zijn zowel op het lichaam als op de vectorruimte, terwijl vermenigvuldiging en deling alleen in het lichaam gedefiniëerd is, en er een operatie scalaire vermenigvuldiging is die het lichaam en de vectorruimte met elkaar in verband brengt). Door deze beperking wordt het formalisme van Strassen wat minder fraai (scalaire vermenigvuldiging op een

vectorruimte wordt ingevoerd door voor iedere scalar een bijbehorende operator in te voeren).

Gegeven een programma als syntactisch object, voert Strassen een tweetal complexiteitsmaten voor dit programma in. Allereerst wordt aan iedere operator een niet negatief gewicht toegekend (de kosten voor deze operator); constanten krijgen gewicht 0. De lengte van een programma ontstaat door de kosten van alle operatoren regelsgewijs op te tellen; de diepte van een programma ontstaat door regelsgewijs de kosten van de operator op te tellen bij het maximum van de kosten van de optredende operanden, en het maximum te vormen van de aldus gevormde regelkosten. De diepte meet dus als het ware de rekentijd op een maximaal parallelle machine terwijl de lengte gewoon de sequentiële rekentijd meet.

Een maat die Strassen buiten beschouwing laat is de breedte van een berekening, of liever gezegd, de breedte van het bijbehorende netwerk. Hierbij tellen we het maximum aantal knopen dat wij op een interne laag bij een zo voordelig mogelijke opsplitsing van het netwerk in opeenvolgende lagen aantreffen. Hierbij dienen wel maatregelen genomen te worden opdat geen kanten meerdere lagen kruisen. In dit verband dient ook als maat vermeld te worden het minimale aantal stenen dat nodig is om de graaf te stenigen (pebble game); de laatste maat meet het aantal geheugenplaatsen dat nodig is om de door het programma verlangde uitdrukkingen te berekenen, waarbij desgewenst hele deelexpressies opnieuw mogen worden uitgerekend. Van deze laatste maat is recentelijk vastgesteld dat de bepaling ervan voor een gegeven graaf een weerzinwekkend lastig probleem kan zijn: het is volledig voor Polynomiaal begrensd geheugen [53].

Ik wil mij in het vervolg van dit verhaal geheel richten op de lengtemaat, d.w.z., wij beperken ons tot het meten van sequentiële tijd voor problemen als matrixvermenigvuldiging. Hetgeen overigens niet wil zeggen dat deze problemen in het kader van parallelle machines geen aandacht zouden krijgen - integendeel (zie bijv. [69,70,78,84,89,100,134,141]).

Merk op dat in de formalisatie van Strassen de kosten van een operator onafhankelijk zijn van de operanden waarop de operator dient te worden uitgevoerd; een vermenigvuldiging van getallen van één bit kost dus even veel als één op getallen van lengte 10000; heeft men hier geen vrede mee, dan moet men overgaan op een model waarbij de algebra niet bestaat uit getallen, maar uit bits. Dit laatste is goed mogelijk en men kan dan ook gemakkelijk onderwerpen als netwerk complexiteit binnen het formalisme brengen. Een verhandeling die zowel parallellisme als eindige precisie in beschouwing neemt is te vinden in [19].

Laat $O = (\underline{op1}, \ldots, \underline{opn})$ de collectie operatoren voor een algebraïsche structuur zijn en zij P een programma met operatoren uit O en gegevens $x_1, \ldots, x_m$, en resultaten $y_1, \ldots, y_n$. Gegeven een algebra V van type O (d.w.z. alle operatoren uit O hebben betekenis als operatie op V) dan berekent het programma P een n-tal functies $f_1, \ldots, f_n$ van V^m naar V. De lengte van P, zeg L(P), vormt een bovengrens voor de lengte van de verzameling functies $f_1, \ldots, f_n$, die ontstaat door L(P) te minimaliseren over alle programmas P die $f_1, \ldots, f_n$ berekenen. De aldus gedefinieerde lengtemaat duiden we aan met $L(f_1, \ldots, f_n)$, waarbij impliciet wordt verondersteld dat de algebra en de bijbehorende kostenmaat voor de operatoren duidelijk is.

Hiermee zijn we helaas nog niet uitgedefiniëerd: van practisch belang is de situatie waarbij men iets wil berekenen terwijl men reeds beschikt over één of meer tussenresultaten; dit effect kan men in het model opnemen door deze tussenresultaten, die zelf op hun beurt functies in de gegevens mogen zijn, als nieuwe constanten, d.w.z. nulplaatsige operatoren, voor de kostprijs nihil aan de algebra toe te voegen, en vervolgens het reeds bekende maatbegrip te hanteren op de nieuwe algebra. Op deze wijze komt men tot een maat $L(f_1, \ldots, f_n/g_1, \ldots, g_k)$. Uiteraard zijn de ordening en eventuele multipliciteiten in de rijen f_i en g_j van geen enkele betekenis; wij mogen de beide argumenten van de maat L(/) opvatten als verzamelingen.

In dit kader komt Strassen tot enkele aardige algemene waarheden: de transiviteitsstelling $L(A/B) \leq L(A/B \cup C) + L(C/B)$ drukt uit dat het berekenen van A uit B via een eventuele omweg C hoogstens extra tijd kan kosten. De eigenschap dat $L(h(A)/h(B)) \leq L(A/B)$ voor een homomorfisme $h : V \to V'$ drukt uit dat voor de berekening van de homomorfe beelden van de gevraagde vormen hoogstens tijdwinst valt te behalen door gebruik van nieuwe mogelijkheden. Zo zelfevident als bovenstaande observaties zijn, ze vormen toch de basis voor de inhoud van [169]. Een dieper resultaat is te vinden in het vervolgartikel Berechnung und Program II [170]; hierin toont Strassen aan dat voor het soort problemen waaraan wij denken (zoals matrixvermenigvuldiging) de lengte van een programma dat van de invoer afhankelijk mag zijn niet wezenlijk korter kan zijn dan die van een programma dat invoer-onafhankelijk is; bij dit bewijs wordt gebruik gemaakt van de taal der schoven en het irreducibiliteitsbegrip uit de algebraïsche meetkunde, en voor een algebraïsch geschoold lezer is het argument in wezen simpel: een invoer-afhankelijk programma is in wezen een samenvoeging van invoer-onafhankelijke programma's die beperkt zijn tot verzamelingen in de verzamelingenalgebra gegenereerd door de Zariski-topologie; als het argumentendomein nu maar irreducibel is

zal een van deze beperkingen op een open verzameling, d.w.z. vrijwel overal, gedefiniëerd zijn.

Met de bovenstaande uiteenzetting heb ik mij overigens meteen schuldig gemaakt aan het hoogdravende algebraïsche taalgebruik dat Strassen dikwijls is verweten door zuivere informatici.

3. MULTIPLICATIEVE COMPLEXITEIT

Bij de bepaling van de lengtecomplexiteit voor het berekenen van functies kan men zich door een handige keuze van de onderliggende algebraïsche structuur en de bijbehorende kostenmaat concentreren op die operaties waarin men geïnteresseerd is. Sommige onderzoekers zijn bijv. geïnteresseerd in minimale aantallen additieve operaties [20,76,77,107,108,135,142,154, 159] en dit kan men onderzoeken doordat men de maat bepaalt met betrekking tot een kostenfunctie waarbij de additieve operaties kosten één hebben, terwijl de overige operaties niets kosten.

Wij zullen ons in dit verhaal verder beperken tot de multiplicatieve complexiteit, waarbij het gaat om vermenigvuldigingen en delingen; zelfs deze formulering is echter niet geheel in overeenstemming met de waarheid want in dit model pleegt men ook de "scalaire" vermenigvuldigingen gratis te leveren. Uitgangspunt is de wiskundige structuur van een k-algebra V, waarbij wij in ons huidige verhaal ervan uitgaan dat k een oneindig lichaam is, zulks ter vermijding van de extra complicaties die optreden als we ook eindige lichamen toelaten. De verzameling V is een ring, of algemener een Abelse groep waarop een bilineaire doch niet noodzakelijk associatieve vermenigvuldiging is gedefinieerd. Tenslotte is er een scalaire vermenigvuldiging van $k \times V \to V$, die voldoet aan de eigenschappen die U welbekend zijn in het geval van een vectorruimte V over k. Eventueel bestaat er in V een deling / als inverse operatie voor de vermenigvuldiging in V. Als voorbeelden mag U denken aan de polynoomring $k[X_1, \ldots, X_n]$, het functielichaam $k(X_1, \ldots, X_n)$ of een machtenreeksenring $k[[X_1, \ldots X_n]]$, maar ook structuren als een lichaamsuitbreiding $k \subset K$, of een algebra als de Quaternionen (over $\mathbb{R}$), of een Lie-algebra over $\mathbb{C}$ vormt een voorbeeld van de bedoelde situatie.

Als kostenfunctie voor de operaties kiezen wij de functie waarin de operaties binnen k, de additieve operaties binnen V en de scalaire vermenigvuldiging (en daarmede impliciet ook het delen van elementen in V door scalairen in k) niets kosten, terwijl de vermenigvuldiging in V en de eventuele deling in V gewicht één hebben. Een consequentie hiervan is dat na het vormen

van een aantal resultaten men voor niets kan beschikken over alle k-lineaire combinaties van deze vormen; men beschikt voortdurend als het ware over een k-lineaire deelruimte van berekende vormen, waarvan de dimensie na iedere "essentiële" operatie met één toeneemt. Dit leidt direct tot een kennelijke ondergrens voor de multiplicatieve complexiteit [172]

$$L(F/G) \geq \underline{\dim}(k(F \cup G)/kG) = \dim(k(F \cup G)) - \underline{\dim}(kG)$$

of, in woorden uitgedrukt: de multiplicatieve complexiteit voor het probleem F te berekenen uit G is minstens de dimensie van het opspansel van de te berekenen vormen relatief het opspansel van de gegevens. Wij zullen deze ondergrens verderop in ons verhaal tegenkomen onder de naam "rijenrang".

In het vervolg zullen wij de elementen van k aanduiden met kleine Griekse letters terwijl Latijnse letters worden gebruikt voor elementen van V; hierbij gaan we ervan uit dat, zolang niet het tegendeel wordt beweerd, verschillende letters elementen in V aanduiden die algebraïsch onafhankelijk zijn; we noemen $a_1, \ldots, a_n$ onafhankelijk over k als voor ieder polynoom F in $k[X_1, \ldots, X_n]$ geldt $F(a_1, \ldots a_n) = 0$ impliceert $F = 0$.

Bij wijze van voorbeeld enkele opmerkelijke eigenschappen van het besproken model. Binnen dit model geldt dat een zekere formalisering van het probleem "polynoomvermenigvuldiging" lineaire complexiteit heeft. De formalisering luidt als volgt: Zij gegeven de polynomen $F = a_0 + a_1X + \ldots + a_nX^n$, $G = b_0 + b_1X + \ldots + b_mX^m$ waarbij de a_i en b_j algebraïsch onafhankelijke elementen uit V zijn. Gevraagd de n+m+1 coëfficienten c_p van het productpolynoom F.G (als elementen in V).

In wezen komt dit neer op de berekening van de volgende bilineaire vormen

$$\begin{aligned}
c_0 &= a_0b_0 \\
c_1 &= a_1b_0 + a_0b_1 \\
c_2 &= a_2b_0 + a_1b_1 + a_0b_2 \\
&\ldots\ldots\ldots \\
c_{n+m} &= a_nb_m.
\end{aligned}$$

De bilineairiteit van dit probleem komt beter tot haar recht door het te formuleren als een "matrix maal vector" probleem:

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ \\ \\ c_{n+m} \end{pmatrix} = \begin{pmatrix} a_0 & & & & & \\ a_1 & a_0 & & & & \\ a_2 & a_1 & a_0 & & & \\ \vdots & & & \ddots & & \\ a_n & & & & a_0 & \\ & \ddots & & & a_1 & \\ & & \ddots & & \vdots & \\ & & & \ddots & a_n & \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Regelrecht uitwerken van de formules leidt tot een programma waarin (n+1)(m+1) essentiële vermenigvuldigingen optreden. Het kan echter ook met n+m+1 vermenigvuldigingen als we het probleem als volgt aanpakken: kies n+m+1 verschillende elementen van k: zeg $\alpha_0, \alpha_1, \ldots, \alpha_{n+m}$. Evalueer de polynomen F en G in deze punten; aangezien de punten zelf en hun machten elementen van het lichaam en derhalve scalairen zijn kosten deze vermenigvuldigingen niets. Vorm vervolgens de n+m+1 producten $p_i := F(\alpha_i)G(\alpha_i)$; kosten n+m+1. Wat rest is een interpolatieprobleem om de coëfficienten van het productpolynoom te berekenen uit de waarden in n+m+1 verschillende punten in k: dit is in wezen het oplossen van het volgende lineaire stelsel:

$$\begin{pmatrix} 1 & \alpha_0 & \alpha_0^2 & \cdots & \alpha_0^{n+m} \\ 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n+m} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha_{n+m} & \alpha_{n+m}^2 & \cdots & \alpha_{n+m}^{n+m} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n+m} \end{pmatrix} = \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n+m} \end{pmatrix}$$

Aangezien de optredende matrix als Vandermonde matrix niet singulier is bestaat zijn inverse; vermenigvuldiging met deze inverse kost niets omdat alle optredende matrixelementen scalairen uit het lichaam zijn.

Een soortgelijk voorbeeld betreft het probleem van het formeel inverteren van een machtreeks: gegeven de machtreeks $\Sigma a_i X^i$ met $a_0 \neq 0$ in k, te bepalen de coëfficiënten van de machtreeks $\Sigma b_i X^i$ zodanig dat $\Sigma a_i X^i . \Sigma b_i X^i = 1$. Omdat hier gevraagd wordt naar de berekening van oneindig veel vormen valt dit probleem niet binnen ons formalisme; wij beperken ons derhalve tot de eerste

k vormen b_i (die kennelijk slechts afhankelijk zijn van de eerste k vormen a_i). Het aldus verkregen probleem blijkt opnieuw bij het naïef nalopen van de formules orde k^2 vermenigvuldigingen te vragen, terwijl Strassen dit verbetert tot orde k.log(k) [172] ook dit is echter niet optimaal - een proces dat de zuivere wiskundige kent als Newton-iteratie, in dit geval ook wel kwadratische Hensel genoemd, stelt ons in staat het probleem te kraken in ongeveer 4k vermenigvuldigingen; zie SIEVEKING [164] of KUNG [85].

In het vervolg van de voordracht zullen wij nog andere voorbeelden tegenkomen, waaronder matrixvermenigvuldiging; dit probleem laat zich thans als volgt formuleren:
te berekenen:

$$\begin{pmatrix} c_{11} & \cdot\cdot & c_{1n} \\ \vdots & & \vdots \\ c_{k1} & \cdot\cdot & c_{kn} \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{km} \end{pmatrix} \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix}$$

Hierbij zijn de a_{ij} en de b_{jp} algebraïsch onafhankelijke elementen van V terwijl de c_{ip} de te berekenen vormen $c_{ip} = \Sigma_j a_{ij} b_{jp}$. Ook in dit voorbeeld zijn de te berekenen vormen bilineaire vormen in de gegevens.

Direct uitwerken van de formules leidt tot een programma dat nmk vermenigvuldigingen vraagt; dat dit niet optimaal is blijkt uit de identiteiten van Strassen die aantonen dat 7 vermenigvuldigingen voldoende zijn voor het geval k = m = n = 2:

$$p_1 = (a_{11}+a_{22})(b_{11}+b_{22}),\ p_2 = (a_{21}+a_{22})b_{11},\ p_3 = a_{11}(b_{12}-b_{22}),$$
$$p_4 = a_{22}(b_{11}+b_{21}),\ p_5 = (a_{11}+a_{12})b_{22},\ p_6 = (-a_{11}+a_{21})(b_{11}+b_{12}),$$
$$p_7 = (a_{12}-a_{22})(b_{21}+b_{22}),$$

waarna geldt:

$$c_{11} = p_1+p_4-p_5+p_7,\ c_{21} = p_2+p_4,\ c_{12} = p_3+p_5,\ c_{22} = p_1+p_3-p_2+p_6.$$

Merk op dat de bovenstaande berekening geen gebruik maakt van de commutativiteit van vermenigvuldiging; alle a_{ij} staan aan de goede kant van de b_{jp}. Het bovenstaande schema is daarom ook correct als de algebra V non-commutatief is, bijv. V is zelf een matrixalgebra. Op deze observatie berust de

verbetering van de orde van complexiteit waartoe Strassen's schema aanleiding geeft.

Vanwege het belang van deze non-commutativiteit zullen we in het vervolg veronderstellen dat wij slechts non-commutatieve algoritmen beschouwen, tenzij anders vermeld; in het laatste geval zullen wij onze complexiteitsmaat L voorzien van een extra index C : $L_C(F/C)$; merk op dat $L_C(F/G) \leq L(F/G)$.

4. DE MATRIXVERMENIGVULDIGINGSEXPONENT

Volgens de standaardmethode is het aantal vermenigvuldigingen voor de vorming van het product van twee n bij n matrices gelijk n^3. We zagen echter dat dit getal n^3 voor n = 2 ongelijk is aan de multiplicatieve complexiteit van dit probleem; die is hoogstens gelijk aan 7. Voeren wij derhalve een functie M in die de multiplicatieve complexiteit van dit probleem meet:

$$M(k,m,n) := L(c_{11}, \ldots, c_{kn} / a_{11}, \ldots, a_{km}, b_{11}, \ldots, b_{mn})$$

waarbij de c_{ip} gegeven zijn door de uitdrukkingen $c_{ip} = \Sigma_j a_{ij} b_{jp}$. We korten de waarde M(n,n,n) af met M(n).

Sinds de ontdekking van Strassen is men geïnteresseerd in de groei van het getal M(n) als functie van n. Omdat wij hierbij immers werken in het non-commutatieve geval kan men het idee om, gegeven een efficient schema voor matrixvermenigvuldiging, dit schema te itereren door de elementen van de matrix zelf matrices te doen zijn, zonder probleem toepassen. Dit leidt tot de volgende stelling:

LEMMA. *Zij* $M(k) \leq q$ *voor zekere* $k > 1$; *dan geldt* $M(k^s) \leq q^s$ *voor iedere natuurlijke* s, *terwijl bovendien* $M(n) \in O(n^\gamma)$ *voor* $\gamma = {}^k\log(q)$.

BEWIJS. De eerste uitspraak volgt met inductie naar s door het volgens aanname bestaande schema dat k bij k matrices vermenigvuldigt in q vermenigvuldigingen te gebruiken voor blokmatrices waarvan de k^{s-1} bij k^{s-1} blokken volgens de inductie-aanname vermenigvuldigd kunnen worden met q^{s-1} vermenigvuldigingen; de kosten van q van deze producten bedragen tesamen niet meer dan q^s.
De tweede bewering laat zich als volgt uit de eerste afleiden: zij $s = \lceil {}^k\log(n) \rceil$; er geldt dat $n \leq k^s \leq nk$; verder is het evident dat $M(n) \leq M(n')$ voor $n \leq n'$.

Hieruit volgt

$$M(n) \leq M(k^s) \leq q^s = k^{s.k\log(q)} \leq (nk)^{k\log(q)} \in O(n^\gamma).$$

Een belangrijkere observatie, die tevens een rechtvaardiging levert om voor het onderhavige probleem te kijken naar het (non-commutatieve) multiplicatieve model is dat een soortgelijke uitspraak geldt voor het totaal aantal arithmetische bewerkingen dat nodig is voor de gevraagde berekening.

LEMMA. *Onder dezelfde aanname als hierboven geldt* $A(n) \in O(n^\gamma)$, *waarbij* $A(n)$ *de volledige complexiteit voor het matrixvermenigvuldigingsprobleem is. Bovendien geldt* $I(n) \in o(n^\gamma)$, *indien* $I(n)$ *de algebraïsche complexiteit voor de inversie van* n *bij* n *matrices voorstelt* [167].

BEWIJS. Stel dat het schema voor k bij k matrices naast de q vermenigvuldigingen ook nog t additieve operaties en scalaire vermenigvuldigingen vraagt. Deze laatste operaties kosten, indien uitgevoerd op m bij m blokmatrices niet meer dan m^2 elementsgewijze operaties van hetzelfde simpele type.

Op grond van deze beschouwingen vinden wij de volgende recurrente relaties voor het benodigde aantal additieve operaties $B(n)$:

$$M(km) \leq q.M(m), \quad B(km) \leq t.m^2 + q.B(m).$$

Met volledige inductie naar s leidt men hieruit af:

$$M(k^s) \leq q^s, \quad B(k^s) \leq t.k^{2s-2} + q.t.k^{2s-4} + \ldots + q^{s-1}.t =$$

$$= t.(q^{s-1} - k^{2s}/q)/(1-k^2/q).$$

Als we nu ook nog gebruik maken van het feit dat $q > k^2$, iets wat we verderop zullen bewijzen, maar waarvoor ook een bewijs reeds is gegeven door STRASSEN [172], dan volgt $B(k^s) \leq \text{Const.}\ q^s$; hiermee volgt het eerste deel van het lemma als tevoren. De uitspraak voor matrixinversie berust op een schema aangegeven door STRASSEN [167], waarin de inversie van een 2m bij 2m matrix wordt herleid tot twee inversies van m bij m matrices, 6 vermenigvuldigingen van m bij m matrices en drie additieve of scalaire operaties.

Een gevolg van het eerste lemma is dat we de volgende limiet mogen definiëren:

$$\gamma := \lim_{n} \log(M(n))/\log(n).$$

Voor dit getal γ geldt dat $M(n) \in O(n^{\gamma+\varepsilon})$ voor iedere $\varepsilon > 0$; of tevens geldt $M(n) \in O(n^{\gamma})$ is een open vraag. Het is wel duidelijk dat zeker niet geldt $M(n) = n^{\gamma}$ voor zekere δ [101].

Op grond van het bovenstaande zou men verwachten dat de multiplicatieve complexiteit van matrix inversie, die wij zullen aanduiden met $H(n)$, van dezelfde orde van grootte is als $M(n)$: in formule $H(n) \in O(M(n))$; is dit inderdaad zo? Bij STRASSEN [172] treffen wij de volgende afschattingen aan:

$H(n+1) \geq H(n)$ (minder triviaal dan het lijkt - een met een één op de hoofddiagonaal uitgebreide matrix met in de n bij n hoofdminor algebraïsch onafhankelijke elementen bestaat immers niet meer uit onafhankelijke elementen en aangezien matrixinversie delingen vraagt zou het kunnen optreden dat bij het aanroepen van het algemene schema voor inversie van n+1 bij n+1 matrices door nul gedeeld wordt);

$M(n) \leq 3H(2n)$; deze grens volgt uit de matrix-identiteiten

$$D(D+I) = (D^{-1} - (D+I)^{-1})^{-1} \text{ en } \begin{pmatrix} 0 & A \\ B & C \end{pmatrix} \begin{pmatrix} I & A \\ B & C+I \end{pmatrix} = \begin{pmatrix} AB & AC+A \\ B+CB & BA+CC+C \end{pmatrix}$$

$H(2n) \leq 2H(n)+6M(n)$ (zie hierboven)

$M(2n) \leq 7M(n)$ (dit is Strassen's algoritme voor het 2 bij 2 geval).

Analyse levert ons een schatting $M(n) \leq 25.H(n)$; weten we bovendien dat $M(n) \leq O(n^{\delta})$ dan volgt tevens $H(n) \leq O(n^{\delta})$, zoals we eerder hebben geschetst, maar het is onbekend of we hier voor δ de matrixvermenigvuldigingsexponent γ zelf mogen invullen; een andere voldoende voorwaarde zou zijn een schatting als $8M(n) \leq M(4n)$ zoals die wordt geïnspireerd door Strassen's observatie dat in een product van blokmatrices

$$\begin{pmatrix} A & 0 & 0 & B \\ 0 & C & D & 0 \\ 0 & E & F & 0 \\ G & 0 & 0 & H \end{pmatrix} \quad \begin{pmatrix} P & 0 & Q & 0 \\ 0 & R & 0 & S \\ T & 0 & U & 0 \\ 0 & V & 0 & W \end{pmatrix}$$

acht kleinere producten zijn terug te vinden.

Op grond van een dergelijke schatting zouden wij, via gebruikmaking van

$$H(2^s) \leq 6M(2^{s-1}) + 12.M(2^{s-2}) + \ldots + 3.2^s.M(1) + 2^s$$

kunnen komen tot een begrenzing $H(2^s) \leq \text{const.}\ M(2^s)$, waaruit zou volgen dat inderdaad $H(n) \in \Theta(M(n))$. Helaas leidt onze observatie over 8 producten die schuil gaan in een vier keer zo groot product niet tot de gezochte schatting, omdat niemand kan garanderen dat acht matrixvermenigvuldigingen acht keer zo duur zijn als één zo'n vermenigvuldiging, zelfs niet als de elementen van de acht paren te vermenigvuldigen matrices niet met elkaar te maken hebben. We zullen verderop dit probleem terugzien in de gedaante van een nog steeds onbewezen vermoeden over de rang van een disjuncte som van tensoren. Dat enig argwaan hier op zijn plaats is werd ingegeven door het omstreeks 1970 ontdekte feit dat de evaluatie van een polynoom in meerdere punten per punt aanmerkelijk goedkoper kan zijn dan polynoomevaluatie in een enkel punt [2,21,22,46].

Tenslotte nog een algemene waarschuwing: de waarde van de matrixvermenigvuldigingsexponent zegt alleen maar iets over het asymptotisch gedrag van de complexiteit. Om vast te stellen voor welke waarde van n een nieuw en efficienter schema ook echt practisch goedkoper wordt is een probleem dat nadere studie vraagt. Hierover zijn de nodige publicaties verschenen, waarin tevens is onderzocht hoe de recursie van Strassen te combineren valt met andere handigheidjes [33,48,49,81,165]. Van alle overige, theoretisch efficientere methoden valt te verwachten dat ze in de practijk onbruikbaar zijn [35,116,118,152]. Ook het verband met numerieke stabiliteit (zij het in een aanmerkelijk verscherpte zin) is onderzocht [24,105,179].

5. OPTIMALE ALGORITMEN VOOR BILINEAIRE EN KWADRATISCHE PROBLEMEN; DE TENSORRANG

Wij zullen ons bij onze beschouwingen verder beperken tot bilineaire en kwadratische problemen. De te berekenen vormen zijn alle homogene vormen van de graad 2 in een polynoomring $K[X_1, \ldots, X_f]$; we spreken dan van een kwadratisch probleem. Is het bovendien mogelijk de verzameling variabelen X_i te splitsen in twee deelverzamelingen $X_1, \ldots, X_n, Y_1, \ldots, Y_m$, zodanig dat ieder monoom in de te berekenen vormen een product $c_{ij}X_iY_j$ is dan spreken we van een bilineair probleem. (In afwijking van onze afspraak zijn de c_{ij} hier lichaamselementen!)

De algemene vorm van een probleem is derhalve:

kwadratisch probleem: bereken $f_p = \sum_{i,j} c_{ijp}X_iX_j$ voor $p = 1, \ldots, k$

bilineair probleem: bereken $f_p = \sum_{i,j} c_{ijp}X_iY_j$ voor $p = 1, \ldots, k$.

Merk op dat bij de formulering van de kwadratische problemen een storende dubbelzinnigheid optreedt. De bijdrage van de term $X_i X_j$ wordt bepaald door de som $c_{ijp} + c_{jip}$; het is dus mogelijk eenzelfde stel vormen op oneindig veel verschillende manieren als probleem te formuleren.

In feite wordt het probleem geheel bepaald door het driedimensionale blok getallen uit het lichaam (c_{ijp}) met $1 \leq i \leq n$, $1 \leq j \leq m$, $1 \leq p \leq k$; een dergelijke driedimensionale matrix noemt men in de algebra een tensor. Tensoren zijn afkomstig uit een vectorruimte die men een tensorproduct noemt. In het onderhavige geval hebben wij te maken met het tensorproduct $K^n \otimes K^m \otimes K^k$, een vectorruimte van dimensie nmk; deze ruimte wordt voortgebracht door elementen van de vorm $x \otimes y \otimes z$ met $x \in K^n$, $y \in K^m$ en $z \in K^k$; met deze elementen mag worden gerekend als ware $\otimes$ een *universele* bilineaire aire operator, d.w.z. er gelden regels zoals $(\alpha x_1 + \beta x_2) \otimes y \otimes z = \alpha(x_1 \otimes y \otimes z) + \beta(x_2 \otimes y \otimes z)$, en deze regels zijn in feite de enige die men mag gebruiken om sommen van tensoren te vereenvoudigen. Gebruikmakende van deze regels is het mogelijk iedere tensor uit te drukken als een lineaire combinatie van tensoren van de vorm $e_i \otimes e_j \otimes e_p$, waarbij de e_i, e_j, e_p basisvectoren voor K^n, K^m respectievelijk K^k zijn. Uiteraard is het zo dat dit tensorproduct in de algebra een diepere functie heeft; het is zelfs mogelijk om in te zien dat een stelsel bilineaire vormen correspondeert met één bilineaire afbeelding $S: K^n \otimes K^m \to K^k$ terwijl dit weer correspondeert met een element van een tensorproduct $\bar{K}^n \otimes \bar{K}^m \otimes K^p$ waarbij de streep boven de vectorruimten aangeeft dat de duale ruimte wordt bedoeld; de geïnteresseerde lezer kan een en ander terugvinden bij BOURBAKI [23]. Het voordeel van een dergelijke beschouwing is overigens duidelijk; men kan algebraïsch inzicht op basisonafhankelijke wijze formuleren, waarbij rekenwerk wordt vervangen door begrip.

Ik noem meteen enkele andere representaties voor het bovenstaande bilineaire probleem:
de trilineaire vorm:

$$F = \sum_p f_p Z_p = \sum_{i,j,p} c_{ijp} X_i Y_j Z_p.$$

Deze trilineaire vorm vertoont een opvallende symmetrie, waarbij de rol van X, Y en Z kan worden gepermuteerd. Deze zelfde symmetrie ligt ook besloten in de weergave als tensor aangezien een tensorproduct op canonieke wijze isomorf is met een soortgelijk product waarin de volgorde van de factoren is verwisseld.

de matrix maal vector schrijfwijze:

$$\begin{pmatrix} f_1 \\ \vdots \\ f_p \end{pmatrix} = \begin{pmatrix} b_{11} & \cdots\cdots & b_{1m} \\ \vdots & & \vdots \\ b_{p1} & \cdots\cdots & b_{pm} \end{pmatrix} \begin{pmatrix} Y_1 \\ \vdots \\ Y_m \end{pmatrix} \qquad \text{met } b_{pj} = \sum_i c_{ijp} X_i$$

een trilineair analogon hiervan:

$$F = (Z_1, \ \ldots \ , Z_p) \begin{pmatrix} b_{11} & \cdots\cdots & b_{1m} \\ \vdots & & \vdots \\ b_{p1} & \cdots\cdots & b_{pm} \end{pmatrix} \begin{pmatrix} Y_1 \\ \vdots \\ Y_m \end{pmatrix} \qquad \text{met } b_{jp} \text{ als boven.}$$

Gebruiken we de laatste representaties, waarbij uiteraard alleen de structuur van de matrix M(X) van belang is, dan kan het aanroepen van de symmetrie waarbij X, Y en Z worden gepermuteerd soms tot verrassende verwantschappen leiden. Beschouw bij wijze van voorbeeld het probleem van de complexe vermenigvuldiging, dat zich laat formuleren via de matrix:

$$\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} X_1 & -X_2 \\ X_2 & X_1 \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} .$$

Verwisseling van X en Z transformeert dit probleem tot

$$\begin{pmatrix} g_1 \\ g_2 \end{pmatrix} = \begin{pmatrix} Z_1 & Z_2 \\ Z_2 & -Z_1 \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}$$

hetgeen op het eerste gezicht een ander probleem is.

Het is interessant de tensor te beschouwen voor het probleem matrixvermenigvuldiging: we moeten hierbij werken met dubbel geïndiceerde variabelen X_{ij} Y_{pq} en Z_{rs}, waarbij we om redenen van esthetische aard de volgorde van de indices bij de Z-variabelen hebben verwisseld; we behandelen dus in feite het probleem $f_{rs} = \sum_j X_{sj} Y_{jr}$. Hierbij geldt $1 \leq s \leq n$, $1 \leq j \leq m$, $1 \leq r \leq k$. De bijbehorende tensor bestaat uit zesvoudig geïndiceerde elementen; wij duiden haar aan met T(n,m,k):

$$c_{pi,qj,rs} = \begin{cases} 1 & \text{als } i = q,\ j = r \text{ en } s = p \\ 0 & \text{anders.} \end{cases}$$

Met gebruikmaking van het Kronecker δ-symbool laat zich dit uitdrukken als

$$c_{pi,qj,rs} = \delta_{iq}\delta_{jr}\delta_{sp}.$$

In deze beschrijving van het probleem krijgt de hiervoor vermelde symmetrie een bijzondere betekenis; verwisseling van de rol van X, Y, en Z correspondeert met permutatie van de drie dimensies n, m en k die het matrixvermenigvuldigingsprobleem beschrijven. Onder deze symmetrie gaat bijv. n bij m maal m bij k over in n bij k maal k bij m; T(n,m,k) gaat over in T(n,k,m).

Om een andere interessante eigenschap van het matrixprobleem te begrijpen hebben wij een operatie nodig die bekend is als het Kronecker product van matrices, doch die zich algemeen laat formuleren voor tensoren. Zij gegeven een tensor $C = (c_{ijp})$ van afmetingen n bij m bij k. Beschouw een andere tensor D van afmetingen n' bij m' bij k' en beschouw een tensor van afmetingen nn' bij mm' bij kk' die verkregen wordt door in C het getal c_{ijp} te vervangen door het c_{ijp}-voud van de tensor D. Deze resulterende tensor duiden we aan met C ⊠ D.

Ook in dit geval geldt dat er een algebraïsche achtergrond is die ik U onthoud. Deze tensor-operatie is in wezen dezelfde als diegene die wij reeds eerder gezien hebben.

De mogelijkheid matrixvermenigvuldiging te zien als vermenigvuldiging van blokmatrices van geschikte afmetingen laar zich nu vertalen tot de volgende simpele formule:

$$T(n,m,k) \boxtimes T(n',m',k') = T(nn',mm',kk').$$

De elementaire verificatie die bestaat uit manipulatie van 12 indices zal ik U besparen.

Men zou mogen verwachten dat de multiplicatieve complexiteit van een probleem iets te maken heeft met een eigenschap van de tensor die het probleem beschrijft. Dit is inderdaad het geval. Om deze eigenschap te achterhalen dienen wij echter eerst na te gaan hoe de optimale algoritmen voor dit type problemen er uit kunnen zien. Dit blijkt in het onderhavige geval geheel te achterhalen te zijn; zie bijv. WINOGRAD [187], STRASSEN [172], of [27,45,64,183].

Allereerst kan men opmerken dat het wellicht goedkoper zou kunnen zijn om voor de berekening van een stel bilineaire vormen over te gaan tot het gebruik van delingen waarbij mogelijkerwijs rationale functies als tussenresultaten ontstaan. Strassen laat zien dat dit niet het geval is [172]. Het bewijs komt erop neer dat men door ontwikkeling van de rationale functies rond een geschikt gekozen punt in de bijbehorende affiene ruimte het verloop van de berekening kan weergeven in een machtreeksenring die de oorspronkelijke polynoomring als deelring omvat. De benodigde translaties kosten niets omdat vermenigvuldiging met lichaamselementen evenals de additieve operaties niets kost. Uiteindelijk zijn wij echter slechts geïnteresseerd in de berekening van een tweedegraads vorm; alle benodigde informatie in de tussenresultaten van de berekening in de machtreeksenring is dus in feite bevat in hun gedeelten van de graad kleiner gelijk 2. Hogeregraads termen zullen, gegeven de eigenschappen van de arithmetiek in machtreeksenringen nooit meer op nuttige wijze kunnen bijdragen tot het eindresultaat; zij dienen integendeel alle tegen soortgelijke termen weg te vallen.

We kunnen de berekening in de machtreeksenring derhalve volgen door stap voor stap het effect na te gaan op de homogene delen van de graad 0, 1 of 2 in de operanden $b_i = b_{0i} + b_{1i} + b_{2i}$ + hogeregraads delen. Hierbij is b_{0i} altijd een element van het lichaam, zodat vermenigvuldigen met of delen door b_{0i} gratis is. Additieve operaties $b_i \pm b_j$ leiden tot additieve operaties op de homogene delen en kosten daarom nog steeds niets; hetzelfde geldt voor scalaire vermenigvuldigingen. Voor het product $b_i.b_j$ geldt:

$$b_i.b_j = b_{0i}.b_{0j} + (b_{1i}.b_{0j} + b_{0i}.b_{1j}) + (b_{2i}.b_{0j} + b_{0i}.b_{2j} + b_{1i}.b_{1j}) + \text{troep}$$

en alleen het product $b_{1i}.b_{1j}$ vereist een vermenigvuldiging die iets kost. Analoog laat een quotient b_i/b_j (waarbij altijd geldt dat $b_{j0} \neq 0$) zich schrijven als $c_0 + c_1 + c_2$ + hogeregraads termen waarbij geldt:

$$c_0 = b_{i0}/b_{j0}, \quad c_1 = (b_{i1} - c_0.b_{j1})/b_{j0}, \quad c_2 = (b_{i2} - c_0.b_{j2} - c_1.b_{j1})/b_{j0}.$$

Ook hier is slechts één product te vormen dat iets kost; het product $c_1.b_{j1}$.

Nu is het zo gesteld dat de homogene gedeelten b_{id} gewoon polynomen uit de polynoomring zijn. Bij het nalopen van de berekening in de machtreeksenring kost iedere multiplicatieve operatie één vermenigvuldiging in de polynoomring, zodat we zonder extra kosten ons van meet af aan hadden kunnen beperken tot berekeningen binnen de oorspronkelijke polynoomring.

Het bewijs heeft echter nog een stuk extra informatie opgeleverd; de enige producten die we moeten vormen hebben de gedaante "lineaire vorm maal lineaire vorm". Het resultaat van een dergelijke vermenigvuldiging is een homogene kwadratische vorm die op zijn beurt nooit zal optreden als operand in enige latere vermenigvuldiging. Dit voert ons tot de uiteindelijke gedaante van algoritmen voor de berekening van bilineaire en kwadratische vormen.

LEMMA. *Zij gegeven een programma dat de vormen* $f_p = \sum_{i,j} c_{ijp} X_i Y_p$ *berekent, dan is het altijd mogelijk dit programma te vervormen tot een programma van de volgende gedaante zonder dat het aantal essentiele vermenigvuldigingen toeneemt:*

vorm q *paar lineaire vormen* $g_1^t = \sum_i \alpha_i^t X_i + \sum_j \beta_j^t Y_j$, $g_2^t = \sum_i \gamma_i^t X_i + \sum_j \delta_j^t Y_j$
(dit kost niets)

vorm q *producten* $h^t = g_1^t . g_2^t$ *(kosten:* q *vermenigvuldigingen)*

Vorm de uiteindelijke resultaten als lineaire combinaties van de h^t : $f_p = \sum_t \eta_p^t h^t$ *(kosten: opnieuw nihil).*

In het geval van kwadratische vormen is er geen sprake van optredende Y's dus alle β en δ zijn = 0. We kunnen nu tevens aangeven wat we bedoelen met een non-commutatieve algoritme. Deze is een algoritme waarbij slechts producten van de vorm $x_i y_j$ worden gevormd, dus all β en alle γ zijn = 0.

Wij zullen van nu af aan commutatieve algoritmen voor bilineaire vormen beschouwen als gewone algoritmen voor kwadratische problemen waarbij het onderscheid tussen de X'en en de Y's is opgeheven.

Het is interessant de samenhang tussen de tensorcoëfficiënten c_{ijp} en de in de algoritme optredende grootheden vast te leggen. Merk op dat de berekende vormen zich laten schrijven als:

$$f_p = \sum_t \eta_p^t h^t = \sum_t \eta_p^t g_1^t . g_2^t = \sum_t \eta_p^t \sum_i \alpha_i^t X_i \sum_j \delta_j^t Y_j = \sum_{i,j} (\sum_t \alpha_i^t \delta_j^t \eta_p^t) X_i Y_j \quad \text{non-commutatief}$$

$$f_p = \sum_t \eta_p^t h^t = \sum_t \eta_p^t g_1^t g_2^t = \sum_t \eta_p^t \sum_i \alpha_i^t X_i \sum_j \gamma_j^t X_j = \sum_{i,j} (\sum_t \alpha_i^t \gamma_j^t \eta_p^t) X_i X_j \quad \text{kwadratisch}$$

Deze schrijfwijze voor het non-commutatieve geval wordt erg mooi bij gebruik van de trilineaire weergave:

$$F = \sum_{i,j,p} (\sum_t \alpha_i^t \delta_j^t \eta_p^t) X_i Y_j Z_p .$$

Kennelijk geldt voor de elementen van de bij het probleem behorende tensor c_{ijp}

$$c_{ijp} = \sum_t \alpha_i^t \delta_j^t \eta_p^t \quad \text{voor het non-commutatieve geval,}$$

$$c_{ijp} + c_{jip} = \sum_t \alpha_i^t \gamma_j^t \eta_p^t + \sum_t \alpha_j^t \gamma_i^t \eta_p^t \quad \text{voor het kwadratische geval.}$$

Deze laatste conditie laat zich ook als volgt formuleren:

$$c_{ijp} - \sum_t \alpha_i^t \gamma_j^t \eta_p^t = -(c_{jip} - \sum_t \alpha_j^t \gamma_i^t \eta_p^t).$$

Noemen we een tensor (a_{ijp}) die voldoet aan $a_{ijp} = -a_{jip}$ voor ieder drietal i,j,p een antisymmetrische tensor dan zien we dat in het kwadratische geval het verschil tussen de gegeven en de berekende vorm wordt weergegeven door een antisymmetrische tensor. Dit klopt, want dit is precies de dubbelzinnigheid in de formulering van een kwadratisch probleem die we aan het begin van deze paragraaf zijn tegen gekomen. Merk op dat het door een antisymmetrische tensor beschreven stelsel kwadratische vormen de triviale nulvorm is.

De bovenstaande formules leveren decomposities van de gegeven tensor (c_{ijp}) als som van q tensoren van een zeer speciale vorm, waarbij het element op de plaats i,j,p gelijk is aan een product van een drietal getallen dat slechts van i,j resp. p afhangt. Een dergelijke tensor noemt men een rang één tensor; zij beschrijft hoe een product gevormd moet worden (via de α_i^t en de δ_j^t resp. γ_j^t) en hoe dit resultaat verdeeld wordt over de diverse te berekenen vormen (via de η_p^t). Wij noteren deze tensor als $a^t \otimes b^t \otimes c^t$, waarbij $a^t = (\alpha_i^t)$, $b^t = (\delta_j^t)$ en $c^t = (\eta_p^t)$.

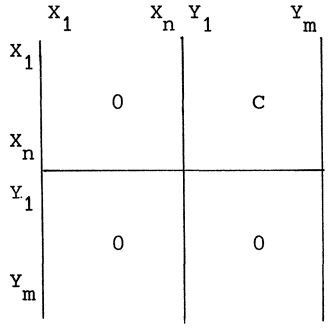
We hebben nu in feite de gevraagde eigenschap van tensoren die de multiplicatieve complexiteit van de bijbehorende bilineaire problemen beschrijft gevonden. Definieer de rang van een tensor als het minimale aantal rang één tensoren waarvoor een decompositie van de gegeven tensor bestaat; in formule:

$$\mu(C) := \underline{\min}\{q \mid C = \sum_{t=1}^{q} a^t \otimes b^t \otimes c^t \text{ voor geschikte } a^t \in K^n,\ b^t \in K^m,\ c^t \in K^k\}.$$

STELLING: *Zij* $C = (c_{ijp})$ *de structuurtensor voor een binilineair probleem* $f_p = \sum_{i,j} c_{ijp} X_i Y_j$. *Dan is de multiplicatieve complexiteit van dit probleem (non-commutatief) gelijk aan de tensorrang van de tensor* C. *De commutatieve multiplicatieve complexiteit voor het corresponderende kwadratische probleem* $f_p = \sum_{i,j} c_{ijp} X_i X_j$ *is gelijk aan de minimale rang van een tensor* C+A *waarbij* A *antisymmetrisch is.*

BEWIJS. De wijze waarop wij de tensor-decompositie hebben afgelezen uit een algoritme toont aan dat de complexiteit naar beneden begrensd is door de vermelde tensorrangen. Omgekeerd laat iedere tensor-decompositie zich rechtstreeks vertalen in een algoritme waaruit de omgekeerde afschatting volgt.

Merk op dat voor het bepalen van de commutatieve complexiteit van een bilineair probleem de bijbehorende tensor moet geacht worden te zijn ingebed in een grotere tensor, gelijk aangegeven in het onderstaande schema:



Wat heeft de algebra ons verder te leren over de tensorrang? Meer in het bijzonder:leert zij ons hoe deze op efficiente wijze kan worden berekend? Het antwoord op de laatste vraag luidt helaas "neen". De algebra leert ons dat voor het "platte" geval k=1, waarbij we te maken hebben met één enkele vorm, resp. een "gewone" matrix, de tensorrang gelijk is aan de gebruikelijke matrixrang, d.w.z. de dimensie van de ruimte opgespannen door de kolom vectoren. Het is een aardige oefening in abstracte begripsvorming na te gaan waarom deze gelijkheid geldt†.

Een andere evidente eigenschap is $\mu(C+D) \leq \mu(C) + \mu(D)$. Deze eigenschap is speciaal interessant voor het geval dat C en D de structuurtensoren zijn van twee problemen over variabelen die niets gemeen hebben. Dit wil zeggen: als $c_{ijp} \neq 0$ en $d_{i'j'p'} \neq 0$ volgt $i \neq i'$, $j \neq j'$ en $k \neq k'$. Het vermoeden bestaat dat in dit geval het gelijkteken geldt [172], maar voorzover mij bekend is dit vermoeden nog open. Het is dit vermoeden dat de lacune vormt in het bewijs van $H(n) \in O(M(n))$ in de voorafgaande paragraaf.

Zoals ik reeds eerder vermeldde kan men de begrippen tensorrang en rang één-tensor onafhankelijk van coördinatenvoorstellingen definiëren. In dit licht kan men beschouwen wat er gebeurt met tensoren onder lineaire afbeeldingen op de vectorruimten waarover de tensoren gedefinieerd zijn. Zij bijv. $C \in U \otimes V \otimes W$ en zij $\phi: U \to U'$, $\psi: V \to V'$ resp. $\chi: W \to W'$ een drietal lineaire afbeeldingen.

†Zie echter GRIGOR'EV [57], voor situaties waarbij deze gelijkheid niet geldt.

Men kan dan definiëren de tensor $\phi \otimes \psi \otimes \chi$ $C \in U' \otimes V' \otimes W'$ en deze heeft een rang die zeker niet groter is dan die van C; zijn de lineaire afbeeldingen isomorfismen dan zijn de rangen uiteraard gelijk.

Voor het onderzoek van de multiplicatieve complexiteit betekent dit dat deze behouden blijft onder basisovergangen in de ruimte der lineaire vormen in de X'en, de Y's en de Z'en. Een basisovergang in de ruimte der Z'en correspondeert met het zoeken naar een ander stel voortbrengers voor het opspansel van de te berekenen vormen, en dat hierdoor de complexiteit niet verandert is een direct gevolg van het gebruik van de multiplicatieve complexiteitsmaat. Transformaties op de ruimte der X'en of de Y's hebben tot gevolg dat de vorm van het probleem er heel anders uit kan gaan zien. Het is zelfs denkbaar dat onder een geschikt gekozen stel isomorfismen de vorm van het bilineaire probleem behouden blijft, terwijl de met een tensordecompositie corresponderende algoritme verandert. Op deze wijze kan men uit een gegeven algoritme een grotere klasse *equivalente* algoritmen destileren. Zie bijv. DE GROOTE [59,60,61], HOWELL & LAFON [67] of HOPCROFT & KERR [64]; men noemt dit de isotropiegroep van de structuurtensor.

Voor het Kronecker-product geldt de schatting $\mu(C \otimes D) \leq \mu(C).\mu(D)$; deze schatting ziet men in door te verifiëren dat het Kronecker-product van twee rang-één tensoren weer een rang-één tensor is. Een gevolg van deze eigenschap dat wij reeds kennen is het effect op de complexiteit van matrixvermenigvuldiging:

$$\mu(T(nn',mm',kk')) \leq \mu(T(n,m,k)) \cdot \mu(T(n',m',k')).$$

Een uit de definitie evidente eigenschap is dat de tensorrang behouden blijft onder permutatie van de factoren van het tensorproduct. Op grond hiervan kunnen we concluderen voor de non-commutatieve complexiteit dat het geen verschil maakt of wij een gegeven probleem onderzoeken, of een van de hieruit afgeleide problemen die verkregen wordt door de tensor te spiegelen (permutaties van X, Y en Z). Toegepast op matrixvermenigvuldiging leidt dit tot de befaamde Symmetriestelling [65,172]:

STELLING. $M(n,m,k) = M(n,k,m) = M(k,n,m) = M(k,m,n) = M(m,n,k) = M(m,k,n)$.

Deze symmetrie gaat verloren voor het commutatieve geval, omdat de rol van de verborgen antisymmetrische tensoren asymmetrisch is. De stelling is voor het commutatieve geval dan ook fout [71].

Een direct gevolg van de symmetriestelling is de volgende schatting voor de matrixvermenigvuldigingsexponent γ:

LEMMA. $M(n,m,k) \leq q$ *impliceert* $\gamma \leq 3.\log(q)/\log(nmk)$.

BEWIJS. $M(nmk) = M(nmk,knm,mkn) \leq M(n,k,m) \cdot M(m,n,k) \cdot M(k,m,n) = (M(n,m,k))^3 \leq q^3$
Pas nu het lemma uit paragraaf 4 toe.

We kunnen ons tenslotte afvragen in hoeverre de tensorrang ons helpt om iets te zeggen over de commutatieve complexiteit. Beschouw daartoe een stel vormen $f_p = \sum_{i,j} c_{ijp} X_i X_j$. Zij $C = (c_{ijp})$ en zij $C^T = (c_{jip})$ de gespiegelde tensor. We weten dat $L(f_1, \ldots, f_k/X_1, \ldots, X_n) = \mu(C+A)$ voor zekere antisymmetrische tensor A. In het commutatieve geval geldt echter nog steeds een symmetrie in de X'en en de Y's: $\mu(C) = \mu(C^T)$. Derhalve geldt [172]

$$2.L(\ldots f_p \ldots / \ldots X_i \ldots) = \mu(C+A) + \mu(C^T+A^T)$$

$$= \mu(C+A) + \mu(C^T-A) \geq \mu(C+C^T+A-A) = \mu(C+C^T).$$

Hieruit volgt voor symmetrische kwadratische vormen ($C = C^T$), zolang de karakteristiek van ons grondlichaam K maar $\neq 2$ is, dat $\frac{1}{2}\mu(C)$ een ondergrens is voor de multiplicatieve commutatieve complexiteit voor kwadratische vormen. Op een gebied waar het gat tussen triviale ondergrenzen en met veel moeite bewezen ondergrenzen op zijn best een factor twee blijkt te zijn deze factor $\frac{1}{2}$ uiteraard een gruwel. De factor $\frac{1}{2}$ wordt in de practijk echter nooit gerealiseerd (JA'JA' [71]).

Een geval waarbij de multiplicatieve complexiteit exact bekend is wordt gegeven door de enkele kwadratische vorm $f = \sum_{i,j} c_{ij} X_i X_j$; deze vorm is, modulo een lineaire transformatie op de ruimte van lineaire vormen in de X_i te schrijven als $f = X_1X_2 + \ldots + X_{2m-1}X_{2m} + \sum_{i=2m+1}^{p} \lambda_i X_i^2$, waarbij de laatste som een definiete kwadratische vorm voorstelt. De getallen m en p zijn bovendien meetkundig gekarakteriseerd; p is de rang van de kwadratische vorm en m is de traagheidsindex. Na deze transformatie is het duidelijk dat $L(f/\ldots X_i \ldots) \leq p-m$, en men kan gemakkelijk nagaan dat de hypothese $L(f/\ldots X_i \ldots) < p-m$ leidt tot een deelruimte van dimensie m waarop de kwadratische vorm identiek nul is, maar dit levert een tegenspraak met de definitie van m als de traagheidsindex, die immers juist de maximale dimensie van een dergelijke nulruimte voorstelt [172].

Ook in dit geval dient men zich te hoeden voor het geval dat de karakteristiek van het lichaam 2 is; dan geldt immers $L(\sum_i x_i^2 / \ .. \ x_i \ .. \ ) = 1$!

In het geval van één bilineaire vorm geldt dat ook de commutatieve en non-commutatieve complexiteit gelijk zijn; dit ziet men in door te kijken naar de vorm van het gesymmetriseerde probleem in het bilineaire geval:

$$C + C^T = \begin{pmatrix} 0 & C \\ C^T & 0 \end{pmatrix},$$

en gebruik makend van de gelijkheid tensorrang = matrixrang.
Joseph Ja'Ja' heeft aangetoond dat deze gelijkheid ook geldt voor een paar bilineaire vormen. In dit geval kan men met gebruikmaking van een klassiek resultaat uit de vorige eeuw (Kronecker-Weierstrass canonieke vorm voor een paar matrices [50,82]) een exacte karakterisering geven voor de non-commutatieve complexiteit, waarbij de resulterende grens zodanig is dat zij bij de vorming van het gesymmetriseerde probleem $C + C^T$ wordt verdubbeld. Zie [55,56,71,72].

6. SIMPELE VOORBEELDEN VAN DECOMPOSITIES

Om in de practijk te kunnen werken met de gelijkheid *non-commutatieve multiplicatieve complexiteit is tensorrang* ware het wenselijk te beschikken over een handzame notatie voor tensoren; tenslotte laten driedimensionale blokken van getallen zich niet of op zijn minst weinig transparant afdrukken op een vel papier. Een veel gebruikte mogelijkheid is de matrix maal vector weergave voor een bilineair probleem. Hierbij is de tensor als het ware plat geslagen door een van de collecties variabelen, bijv. de X'en, te gebruiken als basis voor een stel lineaire vormen die gaan optreden als elementen van de matrix. Hiertoe kan men evenzeer de Y's of de Z'en kiezen, waarbij het behoud van de complexiteit wordt gegarandeerd door de symmetriestelling.
Een rang-één tensor kan men herkennen als een rang-één matrix die in zijn geheel is vermenigvuldigd met een vaste lineaire vorm in de in de matrix optredende variabelen.

Voorbeeld: de tensor T(2,2,2) van 2 bij 2 bij 2 matrixvermenigvuldiging en een decompositie in 7 rang-één tensoren (producten)

$$\begin{pmatrix} X_{11} & X_{12} & 0 & 0 \\ X_{21} & X_{22} & 0 & 0 \\ 0 & 0 & X_{11} & X_{12} \\ 0 & 0 & X_{21} & X_{22} \end{pmatrix} = \underbrace{\begin{pmatrix} X_{11} & X_{11} & 0 & 0 \\ X_{11} & X_{11} & 0 & 0 \\ 0 & 0 & X_{22} & X_{22} \\ 0 & 0 & X_{22} & X_{22} \end{pmatrix}}_{\text{twee producten}} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & X_{22}-X_{11} & X_{11}-X_{22} & 0 \\ 0 & X_{22}-X_{11} & X_{11}-X_{22} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{\text{één product}} +$$

$$+ \underbrace{\begin{pmatrix} 0 & X_{12}-X_{11} & 0 & 0 \\ 0 & 0 & X_{22}-X_{21} & 0 \\ 0 & X_{11}-X_{12} & 0 & 0 \\ 0 & 0 & X_{21}-X_{22} & 0 \end{pmatrix}}_{\text{twee producten}} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ X_{21}-X_{11} & 0 & X_{21}-X_{11} & 0 \\ 0 & X_{12}-X_{22} & 0 & X_{12}-X_{22} \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{\text{twee producten}}.$$

Deze beschrijving, die beoogt Strassen's algoritme begrijpelijk te maken, kan men aantreffen in het proefschrift van FIDUCCIA [43,45] en is naderhand o.m. herontdekt door YUVAL [202].

Andere voorbeelden zijn:

Complexe vermenigvuldiging in drie vermenigvuldigingen:

$$\begin{pmatrix} X_1 & -X_2 \\ X_2 & X_1 \end{pmatrix} = \begin{pmatrix} X_2 & -X_2 \\ X_2 & -X_2 \end{pmatrix} + \begin{pmatrix} X_1-X_2 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & X_1+X_2 \end{pmatrix}.$$

Product van twee lineaire polynomen in drie vermenigvuldigingen:

$$\begin{pmatrix} a & 0 \\ b & a \\ 0 & b \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ b+a & a+b \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} a & 0 \\ -a & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & -b \\ 0 & b \end{pmatrix}.$$

Dit is een symmetrische variant van de decompositie:

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix} = \begin{pmatrix} b & b \\ b & b \end{pmatrix} + \begin{pmatrix} a-b & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & c-b \end{pmatrix}.$$

Quaternionvermenigvuldiging in 8 vermenigvuldigingen: de structuurtensor is

$$\begin{pmatrix} a & b & c & d \\ b & -a & d & -c \\ c & -d & -a & b \\ d & c & -b & -a \end{pmatrix}.$$

Gemakshalve vermenigvuldigen we eerst de eerste rij en daarna de laatste drie kolommen met -1. Dit geeft de tensor

$$\begin{pmatrix} -a & b & c & d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{pmatrix} = \begin{pmatrix} -2a & 0 & 0 & 0 \\ 0 & 0 & -2d & 0 \\ 0 & 0 & 0 & -2b \\ 0 & -2c & 0 & 0 \end{pmatrix} + \begin{pmatrix} a & b & c & d \\ b & a & d & c \\ c & d & a & b \\ d & c & b & a \end{pmatrix}.$$

De eerste tensor is de som van vier rang-één tensoren; de tweede tensor eveneens. Wij herkennen hierin de structuurtensor van de vermenigvuldiging in de groepsalgebra van de viergroep van Klein. Van groepsalgebra's over een commutatieve groep is bekend dat via een isomorfie gedefinieerd via de karakters op de groep, een isomorfisme bestaat met een product van k copieën van het grondlichaam, mits dit lichaam de eenheidswortels ζ_t bevat waarbij t de maximale orde van een element in de groep is. In het onderhavige geval is $t = 2$ dus de eenheidswortel die we nodig hebben $\zeta_2 = -1$ bestaat altijd. Dit leert dat het product te vormen is met vier vermenigvuldigingen; de bijbehorende decompositie van de tensor ziet er als volgt uit:

$$\begin{pmatrix} a & b & c & d \\ b & a & d & c \\ c & d & a & b \\ d & c & b & a \end{pmatrix} = \frac{(a+b+c+d)}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} + \frac{(a+b-c-d)}{4}\begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{pmatrix}$$

$$+ \frac{(a-b+c-d)}{4}\begin{pmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix} + \frac{(a-b-c+d)}{4}\begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

Dit schema duikt op diverse plaatsen in de literatuur op (DOBKIN [37], HOWELL [67], DE GROOTE [58]; zie ook [183]). Eerder had Fiduccia een schema met 10 vermenigvuldigingen beschreven [45] dat ook geldt voor karakteristiek 2; LAFON [91] verspilt een extra tensor om van de viergroep van Klein over te kunnen gaan op de cyclische groep Z_4 en komt aldus op 9 vermenigvuldigingen uit.

Binnen de quaternionenvermenigvuldiging zijn nog enkele interessante problemen terug te vinden zoals het vectoriele product in de driedimensionale ruimte, een combinatie van in en uitproduct, en het simultane product Q_1Q_2 en Q_2Q_1 voor een paar quaternionen Q_1 en Q_2. Zie [39,58,67], waar voor deze problemen decomposities van lengte 5, 6 resp. 10 worden beschreven.

Een verdere generalisatie van de quaternionen vormen de octaven of Caley getallen; we hebben hierbij te maken met een niet associatieve, niet commutatieve algebra, waarvoor een eenvoudig schuiven met plussen en minnen leidt tot een structuurtensor die gelijk is aan die van de groepsalgebra over de Abelse groep $(Z_2)^3$, afgezien van 22 storende mintekens; door hieraan 22 producten te offeren komen we tot een multiplicatieve complexiteit van 30 of minder; zie [46,183].

Bij deze beschouwingen hebben we voortdurend als grondlichaam het lichaam der reële getallen in gedachten. Heffen we deze beperking op dan zien we een omvangrijke klasse van nieuwe voorbeelden: eindig dimensionale lichaamsuitbreidingen. Door een simpele lichaamsuitbreiding $k(\alpha)$ over k van dimensie n te beschouwen als quotient van de polynoomring $k[X]$, volgt gemakkelijk dat vermenigvuldigen in $k(\alpha)$ niet duurder is dan $2n-1$ vermenigvuldigingen in k, zolang k maar minstens $2n-1$ elementen bevat; in de volgende paragraaf zullen we een stelling tegenkomen die vaak aantoont dat deze grens scherp is. Zie ook FIDUCCIA & ZALCSTEIN [46]

Overigens is het van belang rekening te houden met het grondlichaam; als we werken over de complexe getallen blijken twee van de eerder gegeven decomposities niet minimaal te zijn:

Complexe vermenigvuldiging: $\begin{pmatrix} a & -b \\ b & a \end{pmatrix} = \frac{1}{2}(a+ib)\begin{pmatrix} 1 & i \\ -i & 1 \end{pmatrix} + \frac{1}{2}(a-ib)\begin{pmatrix} 1 & -i \\ i & 1 \end{pmatrix}$.

Quaternionenvermenigvuldiging: via de isomorfie van de quaternionenalgebra met de algebra van complexe 2 bij 2 matrices van de vorm $\begin{pmatrix} \overline{z} & w \\ -\overline{w} & z \end{pmatrix}$ mogen we gebruik maken van Strassen's algoritme; 7 complexe vermenigvuldigingen zijn voldoende. Zie bijv. LAFON [91]. Zie verder WINOGRAD [191,193].

Tenslotte dienen in dit verband te worden vermeld het onderzoek naar decomposities van de matrixvermenigvuldigingstensoren T(n,2,k), waarvoor Hopcroft & Kerr aantonen dat ½(3nk + max(n,k)) producten voldoende zijn, en een generalisatie hiervan voor T(n,p,n) met vaste $p \leq {}^{2}\log(n)$, van de hand van BROCKETT & DOBKIN [28] die hiervoor een grens $n^2 + o(n^2)$ afleiden.

Een veel ouder resultaat, dat in dit verband niet onvermeld mag blijven, is de van de commutativiteit gebruik makende algoritme van WINOGRAD [186]. Beschouw opnieuw het probleem met structuurtensor T(n,2,n): te berekenen $f_{ij} = (X_{i1}Y_{1j} + X_{i2}Y_{2j})$, voor $1 \leq i,j \leq n$. Vorm de volgende collecties producten:

$$\begin{aligned} p_{ij} &= (X_{i1} + Y_{2j})(Y_{1j} + X_{i2}) && 1 \leq i,j \leq n \\ q_i &= X_{i1}X_{i2} && 1 \leq i \leq n \\ r_j &= Y_{1j}Y_{2j} && 1 \leq j \leq n. \end{aligned}$$

Dit vergt in het totaal $n^2 + 2n$ producten. Aangezien $f_{ij} = p_{ij} - q_i - r_j$ zijn deze producten voldoende voor de berekening van dit matrixproduct.

Tenslotte twee geïsoleerde records op het terrein van de matrixvermenigvuldiging: LADERMAN [88] toont aan dat M(3) ≤ 23, terwijl SCHACHTEL [145] bewijst dat M(5) ≤ 103; geen van deze resultaten levert overigens een verbetering t.o.v. Strassen voor de matrixvermenigvuldigingsexponent; deze verbetering was voorbehouden aan PAN [116], maar daarover meer in paragraaf 8.

7. ONDERGRENZEN

De karakterisering *non-commutatieve multiplicatieve complexiteit is tensorrang* mag vanuit wiskundig oogpunt buitengemeen aardig en interessant zijn, voor de practisch ingestelde informaticus levert zij amper hulpmiddelen om vast te stellen of een voorliggend schema al dan niet optimaal is. Dit leidt tot het optreden van tergende onzekerheden zoals de vraag *is* M(3) *een priemgetal*? (wij weten slechts dat 19 ≤ M(3) ≤ 23).

Veel aandacht is in de literatuur besteed aan elementaire methoden voor het bewijzen van ondergrenzen voor enerzijds de tensorrang, en anderzijds de commutatieve multiplicatieve complexiteit. De meeste criteria gelden zowel voor het non-commutatieve als het commutatieve geval, maar de mate waarin ze in combinatie mogen worden toegepast is in het non-commutatieve geval aanmerkelijk groter. Daarnaast doet zich het probleem voor dat in het commutatieve geval van de zes denkbare rotaties van de structuurtensor nog

slechts twee overblijven die corresponderen met het onderhavige probleem Het is immers bekend dat de symmetriestelling niet geldt voor het commutatieve geval (JA'JA' [72]).

In feite berusten alle elementaire ondergrenscriteria op een eliminatietechniek die in feite terug gaat op PAN[113]. Gegeven een probleem, wijst men een vermenigvuldiging aan en maakt deze overbodig door hetzij het probleem rechtstreeks te veranderen, hetzij het probleem te beperken tot minder algemene invoer. Deze stap herhaalt men enkele malen, totdat een veel eenvoudiger probleem overblijft, waarvan men een ondergrens kent op basis van een dimensie-argument. De totale ondergrens is de rest-ondergrens, vermeerderd met het aantal succesvol uitgevoerde eliminatiestappen.

Om op de details van de methode in te gaan beschouwen we een bilineair probleem in de gedaante *matrix maal vector*. De matrix bestaat uit lineaire functies in de X_i, terwijl de vector gewoon de kolomvector $(Y_1, \ldots, Y_m)$ is. Door de matrix voor te vermenigvuldigen met de rijvector $(Z_1, \ldots, Z_k)$ ontstaat de aan de tensor ten grondslag liggende trilineaire vorm.

Een algoritme voor dit probleem kan na fatsoenering gebracht worden in de vorm:

$$\begin{pmatrix} f_1 \\ \\ f_k \end{pmatrix} = \begin{pmatrix} \\ M(X) \\ \\ \end{pmatrix} \begin{pmatrix} Y_1 \\ \\ Y_m \end{pmatrix} = \begin{pmatrix} \\ Q \\ \\ \end{pmatrix} \begin{pmatrix} p_1 \\ \\ p_q \end{pmatrix}$$

waarbij Q een matrix van scalairen uit het lichaam K, en $p_1, \ldots p_q$ zijn de door de algoritme gevormde producten. Als tevoren geldt:

$$p_t = (\sum_i \alpha_i^t X_i + \sum_j \beta_j^t Y_j) \cdot (\sum_i \gamma_i^t X_i + \sum_j \delta_j^t Y_j).$$

In het geval van een non-commutatieve algoritme zijn de β_j^t en de γ_i^t alle = 0.

Bij onze beschouwingen zullen we in de inductieveronderstellingen rekening moeten houden met problemen die door het uitvoeren van substituties in eliminatiestappen enigszins zijn verontreinigd. Deze problemen hebben de vorm:

$$\begin{pmatrix} f_1 \\ \vdots \\ f_k \end{pmatrix} = \begin{pmatrix} \\ M(X) \\ \\ \end{pmatrix} \begin{pmatrix} Y_1 \\ \vdots \\ Y_m \end{pmatrix} + \begin{pmatrix} \\ R \\ \\ \end{pmatrix}$$

waarbij de vector R een *residu* is dat bestaat uit ongewenste termen die

alleen bestaan uit hetzij X'en (in het bewijs van het *kolommenrang*criterium), hetzij Y's (in het bewijs van het immuniteitscriterium). Het is duidelijk dat er in het residu geen gemengde termen mogen optreden omdat dan het verschil tussen het bilineaire probleem en de storingstermen verloren gaat.

Verder hebben we voor ons bewijs een nieuw dimensiebegrip nodig, waarmee we de dimensie van platgeslagen tensoren aankunnen. Beschouw een stel vectoren met elementen in $K[X_1, \ldots, X_n] : v_1, \ldots, v_s$; deze vectoren heten K-onafhankelijk als er geen niet triviale K-lineaire combinatie bestaat waarvan de som een constant polynoom voorstelt: in formule

$$\sum_t \lambda_t v_t \in K^m \Rightarrow \lambda_t = 0 \text{ voor iedere } t.$$

Dit afhankelijkheidsbegrip leidt op de gebruikelijke wijze tot een dimensiebegrip en daarmee tot een rangbegrip voor matrices. We hebben in feite te maken met lineaire afhankelijkheid in de vectorruimte $(K[X_1, \ldots, X_n] / K.1)^m$. De lezer zij er echter op bedacht dat niet alle eigenschappen van de gewone matrixrang automatisch geldig blijven. Zo is het niet langer waar dat de rijenrang en de kolommenrang van een matrix gelijk zijn. Een voorbeeld hiervoor wordt gegeven door de rijvector $(X_1, \ldots, X_n)$ die rijenrang 1 en kolommenrang n heeft.

Men kan gemakkelijk nagaan dat de rijenrang in het geval van een bilineair probleem samenvalt met de extra dimensie van het opspansel van de gegevens en resultaten , veroorzaakt door de te berekenen tweedegraads vormen, waarvan wij in paragraaf 3 reeds zagen dat die een ondergrens geeft voor de multiplicatieve complexiteit.

Gemakshalve duiden we de multiplicatieve complexiteit van het onderhavige probleem aan met $L(M)$ resp. $L_C(M,R)$; M stelt hierbij de matrix $M(X)$ voor en R het residu dat alleen op blijkt te treden bij het onderzoek van de commutatieve complexiteit. De rijenrang, resp. kolommenrang van M duiden we aan met $\rho(M)$ resp. $\kappa(M)$.

Het bovengenoemde dimensie-argument laat zich nu als volgt verwoorden:

LEMMA. *(Rijenrang criterium)* $L(M) \geq \rho(M)$; $L_C(M,R) \geq \rho(M)$.

BEWIJS. (Ik geef dit bewijs slechts om te laten zien hoe men dit resultaat kan zien als een eliminatiemethode).
Schrijf $\overline{f} = M(X).\overline{Y} + R = Q.\overline{P}$. Zij $k = p(M)$, de rijenrang van M. Na omnummering van de producten p_t en verwijdering van eventueel afhankelijke rijen uit M (waar-

door het probleem zeker niet moeilijker wordt) mogen we veronderstellen dat de matrix gevormd door de eerste k kolommen van Q (voorzover aanwezig) maximale rang heeft. Deze rang is hoogstens k. Het is derhalve mogelijk een lineaire combinatie van de rijen van Q te vormen, zodanig dat in de resulterende rij de eerste k-1 coëfficiënten nul worden; immers, de k beginstukken van lengte k-1 in de matrix Q zijn afhankelijk. Dit geeft de volgende gelijkheid:

$$(\lambda_1, \ldots, \lambda_k)\left(\begin{pmatrix} & M(X) & \end{pmatrix}\begin{pmatrix} Y_1 \\ \vdots \\ Y_m \end{pmatrix} + \begin{pmatrix} R \end{pmatrix}\right) = (\lambda_1, \ldots, \lambda_k)\begin{pmatrix} & Q & \end{pmatrix}\begin{pmatrix} p_1 \\ \vdots \\ p_t \end{pmatrix} = \sum_{s=k}^{t} \mu_s p_s$$

Op grond van de definitie van rijenrang geldt voor het bovenstaande probleem dat we te maken hebben met een niet triviale tweedegraads vorm, waarvoor de berekening minstens één vermenigvuldiging vraagt. Daarom geldt $t \geq k$, waarmede het bewijs is voltooid.

In het bovenstaande bewijs worden producten p_i verwijderd door over te gaan op een lineaire combinatie van de te berekenen vormen waar deze producten niet langer voor benodigd zijn. Men kan deze stap zien als een *eliminatie in de veranderlijken* Z_p. Het effect van veelvouden van een rij op te tellen bij de andere rijen om daarna deze rij te schrappen had men ook kunnen verkrijgen door in de trilineaire vorm de corresponderende Z_p te vervangen door een lineaire combinatie van de overige Z'en, en daarna het resulterende probleem weer in een nette trilineaire vorm te brengen.

De andere ondergrenscriteria ontstaan door producten te verwijderen door het probleem zodanig te modificeren dat het product nul wordt. Dit bereikt men door in een geschikt gekozen product p_t variabele X_i of Y_j te vervangen door een lineaire combinatie van de overige X'en en/of Y's; als bijv. $\alpha_i^t \neq 0$ en we voeren de substitutie $X_i = -(\sum_{i' \neq i} \alpha_{i'}^t X_{i'} + \sum_j \beta_j^t Y_j)/\alpha_i^t$ uit dan wordt het product p_t gelijk nul en derhalve overbodig.

Deze substitutie heeft het volgende effect op het probleem: in de matrix M(X) ontstaan termen die in het non-commutatieve geval (waarbij alle $\beta_j^t = 0$) gewijzigde lineaire vormen in de X'en zijn waarvan wij niet meer weten dat zij zijn ontstaan uit de gegeven vormen door uitvoering van een substitutie waarvan de coëfficiënten ons alsnog onbekend zijn. In het commutatieve geval treden bovendien lineaire vormen in de Y's op, maar die splitsen we snel af om ze onder te brengen in het residu, waarvan we in dit geval eisen dat het slechts uit vormen in de Y's bestaat.

Hadden we op soortgelijke wijze een substitutie Y_j = ... uitgevoerd, dan ontstaat de vorm in de vector. Het effect hiervan laat zich uitdrukken door een corresponderende operatie op de kolommen van M : tel nader te bepalen veelvouden van kolom j op bij de andere kolommen en schrap kolom j; het residu loopt in het commutatieve geval inmiddels vol met vormen in de X'en.

Hadden we afgezien van het schrappen van kolom j dan was de kolommenrang van de matrix uiteraard ongewijzigd gebleven door deze kolommenoperatie. Het effect van het schrappen van kolom j na deze operatie is derhalve dat de kolommenrang van de matrix hoogstens met één afneemt. Wij verkrijgen op deze wijze het kolommenrangcriterium van WINOGRAD [187].

LEMMA. $L(M) \geq \kappa(M)$; $L_C(M,R) \geq \kappa(M)$ (R *een residu over* $K[\bar{X}]$).

Het bewijs van dit criterium pleegt inductie tot er uiteindelijk nog maar één onafhankelijke kolom over is; het is echter denkbaar dat we in dat stadium nog kunnen profiteren van een restant aan rijenrang. Dit idee ligt ten grondslag aan het *gemengde rang criterium* van FIDUCCIA [43,45]:

LEMMA. *Stel dat de elementen van* M *zodanig zijn dat er geen niet triviale scalairen* $(\alpha_1, \ldots, \alpha_k)$ en $(\beta_1, \ldots, \beta_m)$ *bestaan zodanig dat het polynoom*

$$(\alpha_1, \ldots, \alpha_k) \begin{pmatrix} & \\ & M(X) \\ & \end{pmatrix} \begin{pmatrix} \beta_1 \\ \\ \beta_m \end{pmatrix}$$

een constant polynoom is. Dan geldt $L(M) \geq k+m-1$; $L_C(M,R) \geq k+m-1$ (R *over* $K[\bar{X}]$).

Dit criterium is vaak niet toepasbaar op de gehele matrix M maar wel op een geschikt gekozen grote deelminor.

Keren wij terug tot het effect van een eliminatie van een X_i. Dit geeft aanleiding tot het *rang-immuniteitscriterium* [183], dat van alle elementaire criteria de meest flexibele toepassingen lijkt te bieden. We definiëren dat de matrix M(X) d-immuun is in de variabelen $X_{i_1}, \ldots, X_{i_s}$ indien een willekeurige substitutie van lineaire combinaties van de overige variabelen X_i voor de variabelen $X_{i_1}, \ldots, X_{i_s}$ er allereerst niet toe kan leiden dat een van de overige variabelen niet langer in de matrix M(X) optreedt, en in de tweede plaats de rijenrang van de resulterende matrix na substitutie nog steeds minstens d bedraagt. De eerste conditie dient ervoor om te voorkomen

dat een inductiebewijs stuk loopt op het feit dat een te elimineren variabele ontijdig uit de matrix is verdwenen (merk op dat gegeven de vaste structuur van de vector van Y's dit probleem zich niet voordoet bij het kolommenrangcriterium). Het resultaat laat zich als volgt verwoorden:

LEMMA. *Als* M(X) *d-immuun is in een collectie van* s *variabelen* X_i *geldt* $L(M) \geq d+s$; $L_C(M,R) \geq d+s$ (R *een residu over* $K[\bar{Y}]$).

Op grond van de tegenstrijdige eisen op te leggen aan het residu R blijkt dat het in het commutatieve geval onmogelijk is om het kolommenrangcriterium te combineren met het immuniteitscriterium. In het non-commutatieve geval mag dit wel (zie bijv. BROCKETT & DOBKIN [27]).

Wil men verder komen dan deze criteria leveren, dan rest slechts intensief ploeterwerk. Een eerste uitbreiding is gelegen in het gebruik van lineaire transformaties zonder bijbehorende eliminatie: dit correspondeert met een basisverandering in de X'en (transformatie in de matrix) de Y's (transformatie van de matrixkolommen) of de Z'en (transformatie van de matrixrijen). Deze laatste transformatie kan er toe leiden dat in de matrix Q voor het getransformeerde probleem veel nullen komen te staan, hetgeen wil zeggen dat de vormen die berekend moeten worden zich laten transformeren tot een lineair equivalent stelsel van vormen die ieder voor zich weinig producten vragen. WINOGRAD [196] heeft bijv. opgemerkt dat men een te berekenen vorm die met één product te vormen is altijd mag opnemen in de rij van te vormen producten.

Ik wil nu enkele voorbeelden bespreken.
De tensor van het matrixvermenigvuldigingsprobleem T(n,m,k) laat zich schrijven als:

$$\begin{pmatrix} X_{11} & X_{1m} & & 0 & & \\ X_{n1} & X_{nm} & & & & \\ & & \ddots & & & \\ & & & X_{11} & X_{1m} & \\ & 0 & & & & \\ & & & X_{n1} & X_{nm} & \end{pmatrix}$$

(k blokken langs de diagonaal)

Deze matrix is duidelijk k.n-immuun in de variabelen $X_{12}, \ldots, X_{nm}$, hetgeen leidt tot een ondergrens $M_C(n,m,k) \geq n(k+m-1)$; in het bijzonder $M_C(n) \geq 2n^2-n$. In het non-commutatieve geval kunnen we de drie criteria combineren. Dit is uitgevoerd door BROCKETT & DOBKIN [27]. Elimineer allereerst de onderste kn-n-1 rijen, en vergeet de laatste (k-2)m kolommen.

$$\begin{pmatrix} X_{11} & X_{1m} & & \\ & & \text{rommel} & \\ X_{n1} & X_{nm} & & \\ 0 & 0 & L_1 & L_m \end{pmatrix}$$

waarbij de L_i lineaire vormen in de X_i zijn van de vorm $L_i = X_{1i} + \sum_i$, $\sum_{j=2}^{m} \lambda_{i'j} X_{i'j}$.

We voeren nu twee transformaties uit. Transformatie op de rijen laat de bovenste rij eveneens beginnen met $L_1,\ldots,L_m$, waarna transformatie op de X'en ons in staat stelt in de basis $X_{11}, \ldots, X_{1m}$ uit te ruilen tegen $L_1,\ldots,L_m$. Dit levert de volgende vorm op.

$$\begin{pmatrix} L_1 & \cdots & L_m & & \\ & & & \text{rommel} & \\ X_{n1} & \cdots & X_{nm} & & \\ 0 & & 0 & L_1 & L_m \end{pmatrix}$$

Substitueer vervolgens lineaire vormen in de L_j voor de resterende X_{ij}. Omdat de eerste m kolommen in hun bovenste rij en de laatste m kolommen in hun onderste rij hierdoor niet veranderen heeft het resultaat na afloop van deze substituties nog steeds kolommenrang 2m.

Totaal mogen we het volgende aantal eliminatiestappen tellen:

kn - n - 1 (voor de rijen operaties) + (n-1)m (voor de X'en),

hetgeen leidt tot de ondergrens: $2m + kn - n - 1 + (n-1)m = n(m+k-1) + m-1$. Voor het vierkante geval $n = m = k$ levert dit de grens $M(n) \geq 2n^2 - 1$.*

In het geval n = 2 levert dit een ondergrens 7; in de klasse van non-commutatieve algoritmen is Strassen's methode optimaal. Het is daarnaast bewezen door DE GROOTE [60] dat iedere algoritme voor 2 bij 2 bij 2 matrix-vermenigvuldiging in 7 producten door middel van lineaire transformatie uit Strassen's algoritme te verkrijgen is.

WINOGRAD [189] heeft bewezen dat 7 ook een ondergrens is voor $M_C(2)$. Het bewijs forceert, uitgaande van een schema in 6 vermenigvuldigingen, een lineaire afhankelijkheid tussen de vier productmatrixcoëfficiënten die immers equivalent zouden moeten zijn met een vierdimensionaal stel vormen van rang 3. Voor n = 3 is de ondergrens 19 (PAN [114]) resp. 15 (commutatief), terwijl voor beide problemen 23 de best bekende bovengrens is.

---

*De ondergrens $3n^2-3n+1$ uit de STOC 5 versie van [27] is fout.

Een andere belangrijke toepassing van het immuniteitscriterium (of soms ook het gemengde rangcriterium) is de multiplicatieve complexiteit voor het geval van een nuldelervrije algebra van dimensie n over K. STRASSEN [172] toonde reeds aan dat voor algebra's met 1 een ondergrens n geldt die slechts wordt aangenomen als de algebra isomorf is met een n-voudige macht van het lichaam. FIDUCCIA heeft vermoed [45] en later bewezen [46] (zie ook [183]), dat in een nuldelervrije algebra de ondergrens 2n-1 is. Het bewijs bestaat er uit dat men in de structuurtensor (die dan, aangezien het hier een commutatieve ondergrens betreft, eerst op de juiste wijze dient te worden gespiegeld) na substitutie van alle variabelen door de variabele corresponderende met de eenheid van de algebra X_1, een matrix krijgt die gelijk is aan X_1 vermenigvuldigd met een matrix die het effect beschrijft van vermenigvuldiging met een willekeurig element $\neq 0$ in de algebra. Omdat de algebra geen nuldelers heeft is deze matrix niet singulier, weshalve na vermenigvuldiging met X_1 een matrix met rijenrang n resteert.†

Dit resultaat kan men gebruiken om ondergrenzen te geven voor ingewikkelder algebras [46]. Voor de quaternionen geeft dit een ondergrens 7 voor commutatieve algoritmen. HOWELL & LAFON verbeteren dit in het non-commutatieve geval tot 8 door gebruikmaking van de isotropiegroep van het probleem, die in het geval van een algebra altijd elementen bevat die corresponderen met de eenheden van de algebra. Hierdoor kunnen zij twee producten elimineren door over te gaan op het product van twee vectoriele quaternionen, en voor dit laatste probleem bewijzen zij een ondergrens 6 [67] die overigens ook commutatief geldt [39]. DOBKIN & VAN LEEUWEN bewijzen met handen en voeten een ondergrens 5 (commutatief) voor het vectorproduct in de $\mathbb{R}^3$, terwijl DE GROOTE de (commutatieve) optimaliteit bewijst van zijn schema voor de twee quaternion producten Q_1Q_2 en Q_2Q_1 in 10 producten [58].

Tenslotte nog enkele speciale ondergrenzen uit de matrixvermenigvuldigingshoek. De grenzen M(n,1,1) = n en M(n,m,1) = nm zijn via de elementaire criteria te bewijzen en gelden ook voor het commutatieve geval voor ieder van de spiegelingen. HOPCROFT & KERR [64] tonen aan dat M(2,2,n) = 7n/2; in het commutatieve geval gaat het efficiënter, aangezien Winograd's algoritme leidt tot de grens 2n+2+n = 3n+2. Een combinatie van kolommenrang met rijenrang levert voor dit probleem een commutatieve ondergrens 3n. JA'JA' beschouwt het gespiegelde probleem T(2,n,2) en toont hiervoor een commutatieve ondergrens $M_C(2,n,2) \geq 27n/8$ aan [72]; hiermee is bewezen dat de symmetriestelling in het commutatieve geval niet geldt.

† De meeste algemene vorm van dit resultaat is te vinden in ALDER & STRASSEN [4].

Een speciaal geval betreft matrices van een bijzondere vorm (symmetrisch etc.). Een bekende klasse zijn de Hankel en Toeplitz matrices, waarbij de elementen langs neven- resp. hoofddiagonalen constant zijn. Een dergelijke matrix bevat in het vierkante geval 2n-1 verschillende elementen. Door spiegeling kan men een Hankel matrix overvoeren in een Toeplitz matrix en omgekeerd, dus de multiplicatieve complexiteit voor het probleem een vector te vermenigvuldigen met een dergelijke matrix is voor beide types gelijk. Bekend is dat de grens exact gelijk is aan 2n-1, gesteld dat het lichaam voldoende veel elementen bevat. Vermenigvuldiging van twee $n-1^{e}$ graads polynomen is een speciaal geval. Een volledige beschrijving van de klasse van optimale algoritmen voor dit probleem is te vinden bij WINOGRAD [196]. Een ander speciaal geval is het geval van een circulant met n veranderlijken; over de complexe getallen (of iedere andere ring die de n-e eenheidswortels bevat) is de bijbehorende tensorrang = n. Al deze grenzen voor Hankel- en Toeplitz matrices zijn commutatief geldig. Zie bijv. LAFON [92].

8. DE ONTTRONING EN NEDERGANG VAN 2.81

Zoals ik reeds vermeldde in de inleiding hebben een drietal nieuwe technieken bijgedragen tot de opmerkelijke koersdaling van de matrixvermenigvuldigingsexponent gedurende de afgelopen twee en een half jaar (in tegenstelling tot bij de koers van de dollar is er een garantie; de koers zal nooit beneden de twee dalen)†.

De eerste techniek is de *bundel- en schrap* techniek (Aggregating & Canceling), die is ingevoerd door V. YA. PAN [116,117,118]. Deze techniek vereist het gebruik van vele formules, die zich slechts laten verklaren indien gebruik wordt gemaakt van meerkleurendruk (Pan's uiteenzetting verwees voortdurend naar de optredende groene, rode of purperen termen), en ik beperk mij derhalve bij wijze van illustratie tot een eenvoudig voorbeeld.

Om te begrijpen wat er gebeurt dienen wij ons los te maken van de tot nog toe gehanteerde tweedimensionale representaties voor bilineaire problemen, en te kijken naar de bijbehorende trilineaire vorm. Een tensordecompositie, en derhalve een non-commutatieve algoritme, correspondeert derhalve met een gelijkheid:

$$\sum_{ijp} c_{ijp} X_i Y_j Z_p = \sum_{q} A_q(X) B_q(Y) C_q(Z),$$

waarbij de c_{ijp} lichaamselementen zijn en de A_q, B_q en C_q lineaire vormen.

† of uitstijgen boven de 2.81.

Het aantal termen in de rechtersom heet de lengte van de decompositie en vormt uiteraard een bovengrens voor de complexiteit.

In het onderhavige geval van n bij m bij k matrixvermenigvuldiging waarbij de X'en, Y's en Z'en tweevoudig geïndiceerd zijn, zijn alle c's 0 of 1; de trilineaire vorm laat zich schrijven als:

$$\sum_{ijp} X_{ij}Y_{jp}Z_{pi}.$$

Pan merkt hierbij op dat deze vorm ook te zien is als het spoor van de productmatrix $(X_{ij})(Y_{jp})(Z_{pi})$; op basis van dit inzicht kunnen enkele zaken zoals de symmetriestelling tot elementaire lineaire algebra worden gereduceerd.

Het bundelen bestaat er uit dat een aantal van twee of meer gewenste termen $X_{ij}Y_{jk}Z_{ki}$ tegelijk in één product worden gevormd door meer X'en, Y's en Z'en met elkaar te vermenigvuldigen. Beschouw bijvoorbeeld het geval $n = m = k = 2h$, en laat voor ieder drietal i,j,p het drietal i',j',p' gedefinieerd zijn door $i' = i+1 \pmod n$, $j' = j+1 \pmod n$, $p' = p+1 \pmod n$. Zij tenslotte S de verzameling van drietallen (i,j,p) met i+j+p oneven, Merk op dat $(i,j,p) \to (i',j',p')$ een bijectie levert van S met zijn complement.

Vorm nu de $n^3/2$ producten $(X_{ij} + X_{p'i'})(Y_{jp} + Y_{i'j'})(Z_{pi} + Z_{j'p'}), (i,j,p) \in S$. De som van deze $n^3/2$ producten bevat alle n^3 gewenste termen, maar helaas daarnaast $3n^3$ stoortermen. Deze kunnen wij echter op goedkope wijze kwijtraken aangezien de stoortermen gekarakteriseerd zijn door het optreden van één van de volgende drie types van deelproducten:

$$X_{ij}Y_{i'j''} \quad Y_{jp}X_{j'p''} \quad X_{p'i'}Z_{pi}.$$

Vergaderen wij alle stoortermen met gelijk karakteristiek herkenningspatroon in grotere producten, dan zien wij dat met slechts $3n^2$ extra producten alle stoortermen kunnen worden geschrapt. Ergo $M(n) \leq \frac{1}{2}n^3 + 3n^2$, hetgeen op den duur aanleiding geeft tot matrixexponenten beneden de drie. Helaas geeft de bovenbeschreven grove methode zelf nog geen aanleiding tot verbetering ten opzichte van Strassen.

Nadere analyse levert nog enkele kleinere verbeteringen; men komt tot een bovengrens $\frac{1}{2}n^3 + 9n^2/4$. Om verder te komen moeten we echter iets doen aan de factor $\frac{1}{2}$ bij n^3. Hiertoe gaat Pan over tot de techniek van het trilineaire bundelen , waarbij telkens een drietal gewenste termen wordt gebundeld in een product dat alles bij elkaar 27 termen oplevert (waarvan

24 van minder prettig karakter) volgens het schema:

$$(X_{ij} + X_{j'p'} + X_{p''i''})(Y_{jp} + Y_{p'i'} + Y_{i''j''})(Z_{pi} + Z_{i'j'} + Z_{j''p''}) \quad (i,j,p) \in S$$

voor geschikte S en bijecties ' en ". Van de 24 stoortermen blijken 21 zich te laten samenbundelen in $3n^2$ producten door *karakteristieke* deelproducten aan te geven, maar daarna resten nog een drietal boosdoeners van de vorm $X_{ij}Y_{p'i'}Z_{j''p''}$. Hiermee wordt afgerekend via een speciale schraptechiek.

Het resultaat is een bovengrens $(n^3-4n)/3 + 6n^2$ die met enige moeite wordt verscherpt tot $(n^3-n)/3 + 9n^2/2$. Deze laatste grens levert een optimale exponent voor n = 48: $^{48}\log(47216) = 2.7802$. Verdere verbeteringen via expliciete schema's voor de vermenigvuldiging van volledige matrices zijn sindsdien niet meer verkregen; voor de verdere verlaging van de exponent zouden andere methoden zorg dragen.

9. APPROXIMATIESCHEMA'S

Men kan op de nmk-dimensionale ruimte van alle n bij m bij k tensoren een functie *tensorrang* definiëren, en vervolgens het gedrag van deze functie onderzoeken. Helaas is dit gedrag niet bijster fraai. Het is duidelijk dat de functie discontinuïteiten naar beneden vertoont: tenslotte heeft de triviale nul-tensor van rang 0 niet triviale buren in iedere omgeving en die hebben rang minstens 1; analoog ligt het voor de hand dat de deelvariëteit van rang 1 tensoren wordt benaderd tot op willekeurig kleine afstand door tensoren met rang minstens 2, etc. Op den duur mag men echter verwachten dat de componenten van de ruimte van tensoren met gelijke rang open deelverzamelingen worden die de gehele ruimte gaan overdekken. Dit is in feite een kwestie van voldoende dimensie (of, zoals de fysicalist zou zeggen, *vrijheidsgraden*). Een dergelijke redenering is exact gemaakt door BROCKETT [29].

Het blijkt echter dat zich hierbij een onverwachte tegenslag voordoet; de overdekking met open stukken vertoont *gaten*.

Deze gaten corresponderen met tensoren die willekeurig dicht benaderd worden door tensoren met lagere rang dan zij zelf bezitten. Dit laat zich niet eenvoudig begrijpen, temeer daar dit voor de normale matrixrang zich niet kan voordoen. Waar U aan moet denken is dat een *generieke* tensordecompositie die gedefinieerd is over een van de open componenten gebruik maakt van coëfficiënten die in het algemeen rationale functies in de componenten van de opgesplitste tensor zijn, en het nul worden van een noemer kan leiden

tot het niet langer geldig zijn van zo'n decompositie. Op grond van de geconstateerde *generieke* rang opperde Brockett de mogelijkheid dat de tensoren die corresponderen met matrixvermenigvuldiging best wel eens gelegen zouden kunnen zijn op plaatsen waar de rang een sprong naar boven maakt.

Volgens Schönhage heeft Strassen reeds in 1977 de mogelijkheid gesuggereerd om gebruik te maken van de goedkopere rang van benaderingstensoren, maar het ontbrak destijds aan een goed voorbeeld [152]. Dit voorbeeld is echter einde 1978 gegeven door een viertal Italianen uit Pisa [15], die er zelf enkele maanden voor nodig hadden om de consequenties van hun uitvinding voor de matrixvermenigvuldigingsexponent te zien; toen hun rapport uitkwam [13] moet Pan reeds doende zijn geweest om het approximatie-idee te combineren met zijn eigen technieken [115].

Om het begrip approximatieschema te begrijpen kunnen we zowel topologisch als algebraïsch werken; de laatste beschouwing heeft als voordeel dat zij ook werkt over lichamen waarbij de topologie triviaal is zoals eindige lichamen. De topologische methode blijkt echter tot een lagere grensrang te leiden (zie BINI [14]).

We vervangen bij al onze beschouwingen het lichaam K door een polynoomring K[t], waarbij t een geheel nieuwe variabele is, die de betekenis krijgt van een willekeurig klein te kiezen storing. Gegeven een tensor T en een gelijkheid:

$$T' := \sum_{s=1}^{q} a_q \otimes b_q \otimes c_q \text{ met } a_q \in (K[T])^n,\ b_q \in (K[T])^m,\ c_q \in (K[T])^k.$$

Deze decompositie heet een benaderde decompositie van de lengte q, orde h en (fouten)graad d van de gegeven tensor T indien geldt:

$$t^h T + t^{h+1} S_1 + t^{h+2} S_2 + \ldots + t^{h+d} S_d = T'.$$

Dit betekent in feite het volgende: de decompositie levert een tensor T' waarvan de elementen polynomen in t zijn. Ontwikkelen we T' naar machten van t (waarbij de coëfficiënten nu tensoren zijn) dan vinden we als laagste graads gedeelte de gegeven tensor T vermenigvuldigd met t^h, waarbij h de orde van de decompositie is; daarna volgen nog d hogeregraads termen, waarbij d de foutengraad is. Vullen we voor t elementen $\neq 0$ uit het lichaam in en delen we door t^h dan zien we een rij tensoren van rang q of minder die in de limiet voor $t \to 0$ naar de gegeven tensor T convergeert. T laat zich dus willekeurig dicht benaderen door tensoren van rang q of minder.

We definiëren $\mu_h(T)$ als de minimale lengte van een benaderingsdecompositie van de orde h voor de tensor T. De graad van de decompositie (die we zonder beperking der algemeenheid mogen veronderstellen begrensd te zijn door 2h) doet niet ter zake. Het is duidelijk dat de rij getallen $\mu(T) = \mu_0(T) \geq \mu_1(T) \ldots \mu_h(T) \geq \mu_{h+1}(T)$ een niet stijgende rij natuurlijke getallen is; deze rij heeft derhalve een limiet die we aanduiden met $\overline{\mu}(T)$, en die de algebraïsche grensrang van T heet.

Een alternatief is de topologische grensrang, die gedefinieerd is als de minimale waarde van q zodanig dat T gelegen is in de afsluiting van de collectie van tensoren van rang q of minder. BINI heeft aangetoond dat algebraïsche en topologische grensrang kunnen verschillen [14].

Een voorbeeld moge een en ander verduidelijken. Beschouw de tensor weergegeven door de matrix:

$$\begin{pmatrix} a & 0 \\ b & a \end{pmatrix}.$$

Het probleem is 2 immuun in de variabele b, zodat we een ondergrens van 3 voor de tensorrang kunnen afleiden. De grensrang blijkt echter niet meer dan 2 te zijn, getuige de volgende benaderingsdecompositie van orde en graad 1:

$$a\begin{pmatrix} t & 0 \\ -1 & 0 \end{pmatrix} + (a+tb)\begin{pmatrix} 0 & 0 \\ 1 & t \end{pmatrix} = t\begin{pmatrix} a & 0 \\ b & a \end{pmatrix} + t^2\begin{pmatrix} 0 & 0 \\ 0 & b \end{pmatrix}.$$

Een interessanter voorbeeld is echter het door BINI e.a. [15] aangegeven scherma voor vermanigvuldiging van een onvolledige matrix van de orde 2 met een volle 2 bij 2 matrix. De structuurtensor heeft, na eliminatie van de b-kolommen, rijenrang 4 zodat de tensorrang gelijk 6 is; de grensrang is hoogstens 5 op grond van onderstaande benaderingsdecompositie van graad en orde 1:

$$(ta+b)\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 0 \end{pmatrix} + (ta+c)\begin{pmatrix} 1 & 0 & 0 & 0 \\ t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} - b\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & t & 0 \end{pmatrix} - c\begin{pmatrix} 1 & t & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} +$$

$$+ (b+c)\begin{pmatrix} 0 & t & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & t^2 & t & 0 \end{pmatrix} = t\begin{pmatrix} a & b & 0 & 0 \\ c & 0 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & c & 0 \end{pmatrix} + t^2\begin{pmatrix} 0 & 0 & 0 & 0 \\ a & 0 & 0 & 0 \\ 0 & 0 & 0 & a \\ 0 & b+c & 0 & 0 \end{pmatrix}$$

Het *gat* in de matrix laat zich in dit geval goedkoop verhelpen. Twee keer dit schema toepassen levert een benaderingsdecompositie van lengte 10, orde en graad 1 voor T(3,2,2). Bovendien kunnen we gebruik maken van de volgende multiplicatieve eigenschap van de lengte van benaderingscomposities:

LEMMA. *Het tensorproduct van een decompositie van lengte* q_1, *orde* h_1 *en graad* d_1 *voor een tensor* T_1, *en een decompositie van lengte* q_2, *orde* h_2 *en graad* d_2 *voor een tensor* T_2, *levert een decompositie van lengte* $q_1.q_2$, *orde* h_1+h_2 *en graad* d_1+d_2 *voor* $T_1 \boxtimes T_2$.

We beschikken daarnaast over een symmetriestelling voor de benaderingsrang van de matrixvermenigvuldigingstensor, omdat benaderde decomposities zich laten spiegelen met behoud van lengte, orde en graad. Vormen wij derhalve het product van drie gespiegelde versies van T(3,2,2) dan vinden we een decompositie van de graad en orde 3 en lengte 1000 voor T(12,12,12). Dit geeft op de gebruikelijke wijze aanleiding tot een exponent $^{12}\log(1000)$ = 2.780, zij het slechts voor benaderingsalgoritmen. Het verrassende feit doet zich echter voor dat, op straffe van een factor O(log(n)) in de complexiteit, deze exponent geldig is voor exacte algoritmen, en in de definitie van de matrixvermenigvuldigingsexponent laat deze logaritmische stoorfactor geen sporen na.

LEMMA. *gegeven een benaderingsdecompositie voor de tensor* T *van lengte* q *orde* h *en graad* d; *dan bestaat er een exacte decompositie van lengte* q(d+1) *voor* T.

BEWIJS. Beschouw de decompositie $t^h T + \sum_{s=1}^{d} t^{h+s} S_s = \sum_{r=1}^{q} A_r \boxtimes B_r \boxtimes C_r$ als een polynoomidentiteit. Vullen wij voor t een waarde $\lambda \neq 0$ uit het lichaam in en delen we door λ^h, dan hebben wij in q vermenigvuldigingen een polynoom geevalueerd in het punt λ, waarvan de coëfficienten tensoren zijn, en waarvan de graad gelijk d is. We zijn geïnteresseerd in de constante coëfficiënt, want dat is de tensor T. Door het polynoom te evalueren in d+1 punten λ_i, en daarna te interpoleren (waarbij de non-singulariteit van de corresponderende Vandermonde matrix succes garandeert) kunnen wij derhalve T bepalen. De totale kosten aan producten bedragen (d+1)q. Merk op dat het bewijs het bestaan van minstens d+1 elementen $\neq 0$ in K veronderstelt. Voor eindige K kan de overhead factor $O(d^2)$ worden [152].

Passen wij nu dit lemma toe op een hoge tensoriele macht van een benaderingsdecompositie van een matrixvermenigvuldigingstensor:

STELLING: *Zij gegeven voor een zekere* n,m,k *een decompositie voor* T(n,m,k) *van lengte* q, *orde* h *en graad* d. *Dan geldt voor de matrixvermenigvuldigings-exponent* $\gamma \leq 3 \log(q)/\log(nmk)$.

BEWIJS. Door het vormen van tensoriele machten verkrijgen we benaderings-decomposities van lengte q^{3f}, orde 3fh en graad 3fd voor $T((nmk)^f, (nmk)^f, (nmk)^f)$, en komen zo tot een exacte decompositie van de lengte $q^{3f}(3fd+1)$. Dit geeft een bovengrens $\gamma \leq \log(q^{3f}(3fd+1))/\log((nmk)^f) = (3f.\log(q) + \log(3fd+1))/(f.\log(nmk)) = 3\log(q)/\log(nmk) + \log(3fd+1)/f.\log(nmk))$. Omdat dit voor iedere f een bovengrens voor γ voorstelt, en omdat voor $f \to \infty$ de limiet van de tweede term = 0 is volgt de stelling.

Passen wij dit toe op de decompositie van Bini c.s. dan vinden wij de grens 2.780; het blijkt echter mogelijk op basis van dezelfde benaderings-decompositie de grens $3\ ^6\log(5) = 2.695$ te bewijzen. Dit laatste element hangt samen met de theorie van Schönhage over partiële matrixvermenigvuldiging.

10. ONVOLLEDIGE MATRIXVERMENIGVULDIGING

Het schema voor benaderde matrixvermenigvuldiging van Bini e.a. vertoonde een leemte in de vorm van het ontbreken van een matrixelement, waarvoor 0 was ingevuld. Het bleek weliswaar mogelijk door een ad hoc combinatie te komen tot een volle matrixvermenigvuldigingsalgoritme, maar hiermee verdubbelde zowel het aantal producten als het aantal effectief berekende termen uit de productmatrix, en dat is nadelig voor de verhouding $3\log(q)/\log(nmk)$ die volgens een lemma in paragraaf 5, en de generalisatie tot benaderingsdecomposities in de vorige paragraaf als bovengrens voor de matrixvermenigvuldigingsexponent gehanteerd kan worden.

A. SCHÖNHAGE [152] heeft laten zien dat dit lemma zich rechtstreeks laat generaliseren voor onvolledige matrixvermenigvuldigings schema's. Hiertoe dienen wij dit begrip eerst nader te bepalen.

Veronderstel dat wij een product van een n bij m met een m bij k matrix vormen. Met dit probleem correspondeert een tensor en een trilineaire vorm $\sum_{ijp} X_{ij}Y_{jp}Z_{pi}$. Het aantal termen dat in deze trilineaire vorm optreedt is nmk. We noemen de tensor T' een deeltensor van T(n,m,k) als de bijbehorende trilineaire vorm zich laat schrijven als $\sum_{ijp} c_{ijp}X_{ij}Y_{jp}Z_{pi}$ met $c_{ijp} = 0$ of 1. In de deeltensor hebben we als het ware een aantal termen uit de tensor T(n,m,k) geschrapt. Als het bovendien zo is dat de getallen c_{ijp} te

schrijven zijn als

$$c_{ijp} = e_{ij}.f_{jp}.g_{pi} \text{ met } e_{ij},f_{jp},g_{pi} = 0 \text{ of } 1$$

dan kunnen we aan deze deeltensor op de volgende wijze een betekenis toekennen als onvolledige matrixvermenigvuldiging: $e_{ij} = 0$ betekent dat op de plaats van X_{ij} een nul wordt ingevuld; idem betekent $f_{jp} = 0$ dat Y_{jp} door 0 wordt vervangen. Tenslotte betekent $g_{pi} = 0$ dat we niet geïnteresseerd zijn in het element op plaats i,p in de productmatrix.

Het in manuscript verschenen resultaat van Schönhage beperkt zich tot deeltensoren van deze laatste vorm, waarbij bovendien alle $g_{pi} = 1$; wij zullen in dit geval spreken van een deeltensor in de zin van Schönhage.

Gegeven een deeltensor T' van T(n,m,k) kunnen we het aantal berekende termen bepalen; noem dit f(T') of f als geen verwarring mogelijk is. Voor een deeltensor in de zin van Schönhage laat f zich als volgt bepalen.

Zij voor $1 \leq j \leq m$ het getal $n_j(k_j)$ gedefinieerd als het aantal actieve posities in de j-de kolom van de matrix (X_{ij}) (j-de rij van de matrix (Y_{jp})). De productmatrix ontstaat als de som van alle rang-één matrices die ontstaan door de j-de kolom van de X-matrix te vermenigvuldigen met de j-de rij van de Y-matrix; de bijdrage van deze j-de rang-één tensor is $n_j.k_j$ termen. Kennelijk geldt derhalve dat $f = \sum_j n_j.k_j$.

Het is mogelijk om van deeltensoren op de gebruikelijke wijze tensorproducten of tensoriele machten te vormen. Hierbij worden niet alleen de afmetingen maar tevens het aantal actieve termen vermenigvuldigd: als T' een deel is van T(n,m,k) en U' een deel van T(n',m',k') dan is $T'' = T' \otimes U'$ een deeltensor van T(nn',mm',kk') in de zin van Schönhage, en bovendien geldt $f(T'') = f(T').f(U')$.

In het geval van deeltensoren in de zin van Schönhage kunnen we zelfs nog iets meer zeggen: als $f(T') = \sum_j n_j.k_j$ en $f(U') = \sum_{j'} n_{j'}.k_{j'}$ dan geldt $f(T'') = \sum_{j,j'} n_j n_{j'}.k_j k_{j'}$, waarbij het getal $n_j n_{j'}$ $(k_j k_{j'})$ zich laat interpreteren als het aantal actieve posities in de kolom met index j,j' in de vergrote X-matrix (resp. in de rij met index j,j' in de vergrote Y-matrix). Het aantal actieve termen wordt dus vermenigvuldigd, terwijl een expansie van de schrijfwijze van f als som van producten informatie levert over het patroon van optreden van de actieve posities in het onvolledige matrixvermenigvuldigingsprobleem dat wordt gerepresenteerd door de producttensor.

Ter illustratie het patroon van de linker matrix in de derde tensoriele macht van het onvolledige matrixvermenigvuldigingsprobleem uit het schema van Bini e.a.:

$$\begin{pmatrix} X & X & X & X & X & X & X & X \\ X & 0 & X & 0 & X & 0 & X & 0 \\ X & X & 0 & 0 & X & X & 0 & 0 \\ X & 0 & 0 & 0 & X & 0 & 0 & 0 \\ X & X & X & X & 0 & 0 & 0 & 0 \\ X & 0 & X & 0 & 0 & 0 & 0 & 0 \\ X & X & 0 & 0 & 0 & 0 & 0 & 0 \\ X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{vgl.: } (2+1)^3 = 8 + 3.4 + 3.2 + 1$$

Wij zijn nu in staat om Schönhage's hoofdstelling te formuleren:

STELLING. *Zij* T' *een deeltensor in de zin van Schonhage van* T(n,m,k) *met als aantal actieve termen* $f = \sum_j n_j.k_j$; *zij gegeven een benaderde decompositie van* T' *van de lengte* q, *orde* h *en graad* d; *dan wordt de matrixvermenigvuldigingsexponent begrensd door* $\gamma \leq 3.\log(q)/\log(f)$.

Voor het geval van het schema van Bini e.a. levert dit de beloofde bovengrens 3.log(5)/log(6) = 2.695. In het geval van volle matrices geldt f = nmk en dan ontstaat de in de vorige paragraaf bewezen stelling.

Ik geeft een schets van het bewijs van deze stelling. Zij s een nader te kiezen getal. Vorm de s-voudige tensoriele macht van T' en noem deze $T^{(s)}$. Dit is een deeltensor van $T(n^s,m^s,k^s)$ in de zin van Schönhage, waarvoor enerzijds een benaderde decompositie bestaat van de lengte q^s, orde sh en graad sd, terwijl anderzijds het patroon van de actieve posities in de X- resp. Y-matrix kan worden afgelezen uit de expansie van $(n_1 + \ldots + n_m)^s$ resp. $(k_1 + \ldots + k_m)^s$. Hiervoor gebruiken we de multinomiale ontwikkeling:

$$(n_1 + \ldots + n_m)^s = \sum_{s_1+\ldots+s_m=s} \binom{s}{s_1, \ldots, s_m} n_1^{s_1}. \ldots .n_m^{s_m} .$$

Een analoge ontwikkeling geldt voor $(k_1 + \ldots + k_m)^s$.

Kies een willekeurig m-tuppel $s_1, \ldots, s_m$ met $s_1 + \ldots + s_m = s$ uit deze expansie. De expansie leert ons dat er $M := \binom{s}{s_1, \ldots, s_m}$ indices j" = $j_1, \ldots, j_s$ zijn waarvoor de j"-de kolom uit de X-matrix (resp. de j"-de rij uit de Y-matrix) $N := n_1^{s_1}. \ldots .n_m^{s_m}$, (resp. $K := k_1^{s_1}. \ldots .k_m^{s_m}$) actieve posities bevat. We vervaardigen nu een nieuwe deeltensor van $T(n^s,M,k^s)$ door alle andere kolommen in de X-matrix, resp. alle andere rijen uit de Y-matrix te schrappen. Noem deze deeltensor T^*; het is gemakkelijk in te zien dat de benaderde decompositie voor $T^{(s)}$ na het invullen van de nodige nullen op plaatsen van actieve variabelen kan worden gebruikt als decompositie voor T^*. De tensor T^* is een deeltensor in de zin van Schönhage, waarbij iedere kolom van de X-matrix N actieve posities bevat en iedere rij van

de Y-matrix K actieve posities vertoont. Helaas staan deze actieve posities per kolom in andere rijen en omgekeerd.

Schönhage gaat vervolgens de deeltensor $T^{\%}$ comprimeren. Neem aan dat het lichaam voldoende veel verschillende elementen bevat om de volgende Vandermonde-achtige matrices te kunnen vormen:

$$G = \begin{pmatrix} 1 & \cdots & 1 \\ \alpha_1 & & \alpha_{n^s} \\ \vdots & & \vdots \\ \alpha_1^{N-1} & \cdots & \alpha_{n^s}^{N-1} \end{pmatrix}$$

afmetingen M bij n^s met alle M bij M minoren niet singulier

Q : een soortgelijke matrix van afmetingen K bij k^s met non-singuliere K bij K minoren.

Beschouw nu het product $G.(X_{ij})(Y_{jp}).Q$; hiervoor bestaat nog steeds een decompositie van lengte hoogstens q^s, orde sh, graad sd; de matrix X" = $G.(X_{ij})$ heeft afmetingen N bij M terwijl Y" = $(Y_{jp}).Q$ afmetingen M bij K heeft. De j-de kolom uit X" bestaat uit lineaire vormen in de M optredende variabelen X_{ij} in de j-de kolom van de X-matrix. Analoog voor de j-de rij van Y". De matrix die beschrijft hoe deze lineaire combinaties gevormd worden is echter een M bij M minor van G die per kolom verschillend kan zijn; volgens de constructie was deze minor niet singulier. Op grond hiervan kunnen wij in een basis van de lineaire vormen de optredende X_{ij} elementen uit de j-de kolom uitwisselen tegen de vormen in de j-de kolom van X"; doen wij dit voor alle kolommen tegelijk en idem voor de rijen van de Y-matrix en Y", dan komen we tot de aangename bevinding dat we onze gegeven decompositie hebben getransformeerd tot een benaderingsdecompositie voor de volledige matrixvermenigvuldigingstensor T(N,M,K).

Op grond van de stelling uit de vorige paragraaf vinden we een bovengrens $\gamma \leq 3.\log(q^s)/\log(NMK)$. Het is dus nodig het product P := NMK groot te maken. Wij bereiken dit door s = r.f te kiezen; voor deze keuze van s wordt het product P = NMK maximaal als we kiezen $s_j = r.n_j k_j$ zoals een simpel *schuif en verbeter* argument laat zien. De multinomiaalcoëfficiënten worden afgeschat met het onderstaande gevolg van STIRLING [36]: $\log(x!) = (x+\frac{1}{2})\log(x) - x + \theta$, $0 < \theta \leq 1$. Het resultaat is de volgende ondergrens voor P:

$$\log(P) \geq r.f.\log(f) - m(1+\tfrac{1}{2}\log(rf)),$$

zodat voor de matrixvermenigvuldigingsexponent een bovengrens ontstaat:

$$\gamma \leq 3.s.\log(q)/(s.\log(f) - m(1+\tfrac{1}{2}\log(s))) = 3.\log(q)/\log(f) + o(1) \text{ voor } s.$$

Hiermede is het bewijs voltooid.

De oplettende lezer kan hieruit concluderen dat Bini c.s. te vroeg hun schema hebben ingepakt tot een schema voor volle matrixvermenigvuldiging; ze hadden er eerst een 999-e tensoriele macht van moeten vormen en daarna een handig verpakkingsschema moeten hantęren. Het is duidelijk dat dit tot efficiëntere schema's leidt voor matrixformaten waarvan wij slechts nachtmerries kunnen hebben.

Schönhage verkrijgt in [152] een verbetering van de exponent tot 2.609 op basis van patronen die hij expliciet aangeeft met bijbehorende benaderde decompositie; de 2.609 ontstaat als 3 log(15)/log(26) in een familie benaderde decomposities van lengte rs + 1, orde 2 en graad 1 voor het onvolledige matrixvermenigvuldigingsprobleem met 2 + 2rs - r - s optredende termen volgens het patroon:

$$\begin{pmatrix} X & X & X & \dots & X \\ 0 & X & X & \dots & X \\ & & & & \\ 0 & X & X & \dots & X \end{pmatrix} \begin{pmatrix} X & X & X & \dots & X \\ X & 0 & 0 & \dots & 0 \\ & & & & \\ X & 0 & 0 & \dots & 0 \end{pmatrix}$$

r bij s+1 s+1 bij 1+(r-1)(s-1)

waarbij r = s = 4 het beste resultaat blijkt te geven. De bijbehorende benaderde decompositie spaar ik U; wellicht een aardige opgave voor een winteravond.

11. EPILOOG

Op het 6e ICALP congres te Graz hield Schönhage een voordracht over storage modification machines , maar de enige vraag uit het publiek was naar de op dat moment best bekende waarde voor γ : resultaat 2.54. Drie maanden later, van 21 tot 28 oktober 1979 organiseerden V. Strassen, A. Schönhage & C.P. Schnorr voor de vierde keer een Oberwolfach Tagung over Komplexitätstheorie. Algebraïsche complexiteit was het centrale thema op deze Tagung, en vele experts waren verzameld. De Italianen waren vertegenwoordigd door

F. Romani en uit de VS waren Winograd & Pan (tegenwoordig woonachtig in de VS) overgekomen. Grigor'ev uit Leningrad zond een abstract maar was verhinderd om te komen. Ook M. Atkinson uit Cardiff en P. Schuster uit Tübingen hadden bijgedragen op dit terrein. Op donderdagavond was er een extra bijeenkomst gewijd aan matrixvermenigvuldiging. De samenvattingen zijn gepubliceerd in het EATCS bulletin [160].

Op vrijdagavond werd door Winograd & Pan een nieuw schema verzonnen dat drie parameters bevatte; één van de aanwezigen wist de mini-computer van Oberwolfach aan de praat te krijgen om de resulterende exponenten uit te rekenen, zodat, terwijl andere deelnemers de bibliotheek bezochten of een fles wijn uit één van de alom aanwezige koelkasten aan het nemen waren, de beide ontdekkers van het nieuwe schema voor een geschikt stel waarden van de parameters het getal 2.521812716 op het scherm zagen passeren. Waarvan acte in het gastenboek.

Voor zover mij bekend is dit resultaat nog ongepubliceerd.† De enige officiële vermelding staat in een recent artikel van ROMANI [139]. Het afgelopen jaar circuleerden daarnaast geruchten over een verbetering tot 2.51 en zelfs 2.49 is genoemd; H. Wozniakowski meldde mij recentelijk dat deze geruchten inmiddels achterhaald zouden zijn.† Een ander gerucht dat ik heb opgevangen is dat het vermoeden dat de rang van de som van twee disjuncte tensoren gelijk is aan de som van de beide rangen bewezen zou zijn; ook hiervoor ontbreekt het mij aan nadere informatie. Wellicht slaat het gerucht op het speciale geval dat is opgelost in [12].

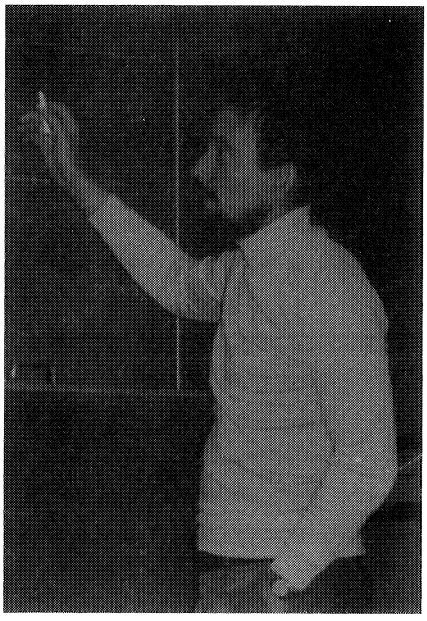
Het gebied is duidelijk nog steeds in beweging. Begin 1981 (1 tot 8 februari 1981) was er weer een Tagung in Oberwolfach, maar het accent lag daar minder op de Algebraïsche Complexiteitstheorie. De Algebraïci kwamen weer bijeen in november 1981.

Tenslotte een viertal portretten van de belangrijkste *dramatis personae* die ik U niet wil onthouden.

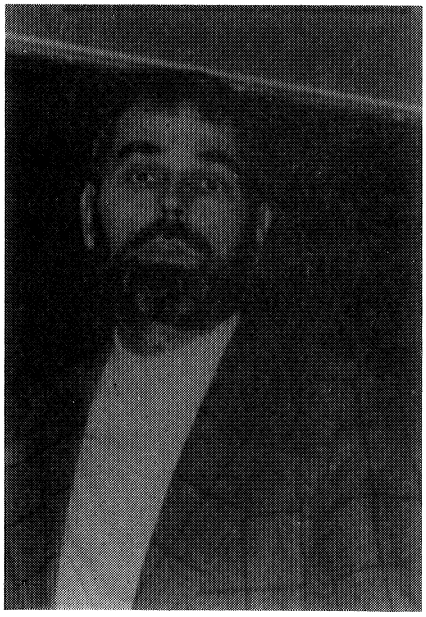
---

† Inmiddels eveneens gepubliceerd in [35,80,117]. De achtergrond van de genoemde geruchten wordt verklaard in de volgende paragraaf.

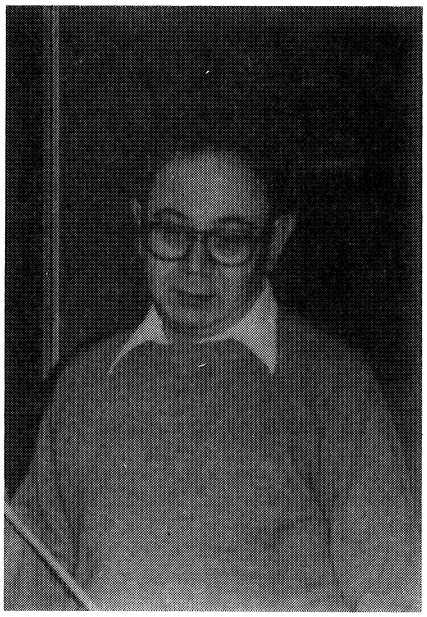
V.Ya. Pan



A. Schönhage



V. Strassen



S. Winograd

12. ONTWIKKELINGEN SINDS 1980

Het zal, gegeven de voorgeschiedenis duidelijk zijn dat de genoemde grens van 2.521812716.... niet lang stand zou houden. Achteraf blijkt het zo te zijn dat het Schönhage† was die via aan extra handigheidje met het Pan-Winograd schema de grens omlaag wist te halen tot 2.521801..... (zie LAZARD [98] - Lazard was een van de deelnemers aan de bijeenkomst te Oberwolwach Oct. 1979, en zijn indrukken hebben eveneens mogen leiden tot een overzichts-artikel). Volgens Lazard heeft nadien Pan een verbetering tot 2.5161 geclaimd, en nadien zijn de geruchten omtrend een verbetering tot beneden de 2.5 gaan circuleren. Uiteindelijk is een betrouwbare verbetering verschenen in het rapport van COPPERSMITH & WINOGRAD [35]. Dit rapport levert naast een verbetering tot 2.495548 de bevestiging van een reeds lang levend vermoeden - de matrix exponent is slechts een limiet. Coppersmith & Winograd tonen in feite aan dat iedere benaderde decompositie van een (onvolledige) matrixvermenigvuldigingstensor kan worden gebruikt om aan andere decompositie te maken voor een grotere tensor die aanleiding geeft tot een lagere waarde voor de exponent: *Everything you can do I can do better*!

De bijdrage van Coppersmith & Winograd kan worden beschouwd als een nieuwe techniek, die een constructie als aangegeven door Schönhage tijdens zijn voordracht te Oberwolfach Oct. 1979 generaliseert. Beschouw bij wijze van voorbeeld een exacte decompositie van $T(m,n,p)$ van lengte $L = M(m,n,p)$. Coppersmith & Winograd tonen aan dat op basis van deze decompositie een benaderde decompositie van een som-tensor $T(m,n,p) \oplus T(1,r,1)$ geconstrueerd kan worden van lengte $L' = L+n$, waarbij $r = L-n(m+p-1)$. De benaderde decompositie wordt verkregen door aan de gegeven decompositie van $T(m,n,p)$ op listige wijze extra termen toe te voegen die allen de variabele t bevatten; de tensor $T(m,n,p)$ is derhalve nog steeds te schrijven als som van de desbetreffende rang één tensoren, zij het nu als benaderde schrijfwijze. Voegen we echter een n-tal extra geschikt gekozen rang één tensoren toe dan blijken alle stoortermen plotseling hoofd term te zijn geworden en hebben we de tensor $T(1,r,1)$ voor slechts n vermenigvuldigingen erbij gerekend.

Hierna is het een kwestie van eenvoudige analyse om aan te tonen dat deze constructie bij machte is ieder bestaand schema te verbeteren. Neem gemakshalve aan dat $n=m=p$. Dan geldt $r = L-2n^2+n$. Zij γ_n de bovengrens voor de matrixvermenigvuldigingsexponent verkregen uit het bovenstaande schema $\gamma_n = {}^n\log(L)$; bekend is dat voor ieder natuurlijk getal s geldt

---

† Zie ook de SICOMP versie van [152]

$\gamma_{n^s} \leq {}^{n^s}\log(L^s)$. Passen we het resultaat van Coppersmith & Winograd toe dan vinden we een nieuw benaderingsschema voor $T(n^s,n^s,n^s) \oplus T(1,r_s,1)$ met lengte $L^s + n^s$, waarbij $r_s = L^s - 2_n^{2s} + n^s$. Dit benaderingsschema geeft volgens het resultaat van Schönhage een bovengrens voor γ :

$$\gamma \leq 3.\log(L^s+n^s)/\log(n^{3s}+L^s-2n^{2s}+n^s) =$$

$$= 3.\log(n^{s\gamma_n}+n^s)/\log(n^{3s}+n^{s\gamma_n}-2n^{2s}+n^s)$$

aangezien $2 < \gamma_n < 3$ mogen wij voor voldoend grote s de noemer afschatten met $\log(n^{3s}+\frac{1}{2}n^{s\gamma_n})$, zodat wij vinden:

$$\gamma < 3.\log(n^{s\gamma_n}(1+n^{s(1-\gamma_n)}))/\log(n^{3s}(1+\tfrac{1}{2}n^{s(\gamma_n-3)})) \ ;$$

Gebruik makende van $\frac{1}{2}x < \log(1+x) < x$ voor $x \downarrow 0$ leiden we hieruit af

$$\gamma < \frac{3.s\gamma_n\log(n)\ (1+n^{s(1-\gamma_n)}/s\gamma_n\log(n)\ )}{3.s\ \log(n)\ (1+n^{s(\gamma_n-3)}/12.s\log(n))} = \gamma_n.\ U_s < \gamma_n$$

want het quotient U_s is voor voldoend grote s kleiner dan 1 op grond van $1 - \gamma_n < \gamma_n - 3$.

De bovenstaande berekening toont aan dat het schema dat aanleiding gaf tot de grens γ_n niet optimaal was. Wellicht zal de oplettende lezer tegenwerpen dat het resultaat van Schönhage slechts asymptotisch tot verbetering leidt, waarbij geen rekening is gehouden met constanten en/of termen van de orde log(n). Dit bezwaar is gegrond. Een poging om het bovenstaande bewijs te generaliseren tot een bewijs voor de bewering "$M(n) \leq C.n^{\alpha} \Rightarrow \gamma < \alpha$" is tot mislukken gedoemd. Deze sterkere bewering zult U in [35] dan ook niet aantreffen. De ervaring leert dat velen het resultaat uit [35] echter op de gesuggereerde wijze misverstaan. Om na te gaan hoegroot de verbetering is dient men naar expliciete voorbeelden van schemas en de verbeteringen er op te kijken. Op deze wijze komen Coppersmith & Winograd (na drie iteraties) op de genoemde grens van 2.495548; verder iteren geeft slechts verbetering in niet afgedrukte decimalen. Wellicht komt men op basis van weer andere schemas tot verdere verbeteringen.

De literatuurlijst is, in vergelijking met de lijst uitgedeeld op het colloquium, uitgebreid met 26 titels. Naast het bovenbesproken artikel van Coppersmith & Winograd dient de tweede editie van Knuth vol 2 te worden genoemd; hierin is de theorie tot en met de 2.52... van Pan & Winograd herleid

tot enkele vraagstukken [80] . Ik wil ook wijzen op het inmiddels verschenen artikel van GRIGOR'EV [57] , waaruit blijkt dat het "tensorrang is matrixrang" uitgangspunt niet geldt over willekeurige algebraische structuren. Andere bijdragen aan de uitbreiding zijn artikelen die in eerdere instantie over het hoofd waren gezien, zoals de bijdragen van ATKINSON e.a. [8,9,10,11]. Tenslotte bereikte mij tijdens de correctie van dit manuscript de definitieve versie van SCHÖNHAGE [152] .

De lange duur van de correctie stelt mij ook nog in staat enkele opmerkingen te maken over de Obewolfach bijeenkomst van Nov. 01-07 1981. Winograd deed uitgebreid verslag over [35] . De heer Lichteig (Konstantz) onderzocht de meetkundige dimensie van de ruimten $\{\tau \mid \mu(\tau) \leq q\}$ voor tensoren $\tau \in K^n \oplus K^m \oplus K^k$, en zit dus weer op het spoor van BROCKETT [29] . Bini beschouwde commutatieve benaderingsschemas's, die niet bruikbaar zijn als uitgangspunt voor een theorie als die van SCHÖNHAGE [152] . Voor dit soort schema's kan men wel een "exponent" defineren, en de vraag is vervolgens of deze soms kleiner dan γ kan zijn. H.F. de Groote houdt zich bezig met de karakterisering van die algebra's waarvoor de ondergrens van ALDER & STRASSEN [4] scherp is. Alder zelf vertelde dat over een algebraisch afgesloten lichaam algebraische en topologische grensrang gelijk zijn. Er waren uiteraard veel meer voordrachten over onderwerpen buiten het bereik van deze syllabus. In tegenstelling tot 2 jaar eerder mochten wij echter geen nieuwe bovengrens voor γ inhuldigen.

13. LITERATUUR

Met opzet heb ik de literatuurlijst niet beperkt tot het enge gebied van de bilineaire en kwadratische complexiteit zoals dit in deze voordracht aan de orde is gekomen. Bij het doorspitten van recente jaargangen van diverse tijdschriften en congresverslagen, heb ik ook titels opgenomen over andere onderwerpen uit de algebraïsche complexiteitstheorie; additieve complexiteit, problemen over polynoomevaluatie, Boolse complexiteit, Algebraïsche complexiteit voor parallelle machines of nog algemenere modellen komen aan de orde. Voor congresbijdragen die later als tijdschriftartikel zijn verschenen is een korte verwijzing naar de congresproceedings vermeld bij de opgave van het tijdschriftartikel, om wille van de datering. Referenties van opgenomen artikelen zijn niet nagetrokken; de lijst zal dus verre van volledig zijn.

Gebruikte afkortingen voor frequent optredende congressen en tijdschriften:

ICALP : International Colloquium on Automata Languages and Programming
MFCS : Mathematical Foundations of Computer Science
FCT : Fundamentals of Computation Theory
STOC : (ACM- SIGACT) Symposium Theory of Computing
FOCS : (IEEE) Foundations of Computer Science Symposium (tot 1975 genaamd: Symposium on Switching and Automata Theory (SWAT))
JACM : Journal of the Association of Computing Machinery
SICOMP : Siam Journal on Computing
IPL : Information Processing Letters
TCS : Theoretical Computer Science
LAA : Linear Algebra and Applications

[1] ALDEMAN, L., K.S. BOOTH, F.P. PREPARATA & W.L. RUZZO, *Improved time and space bounds for Boolean matrix multiplication*, Acta Informatica 11 (1978) 61-75.

[2] AHO, A.V., J.E. HOPCROFT & J.D. ULLMAN, *The design and analysis of computer algorithms*, Addison Wesley 1974.

[3] AHO, A.V., K. STEIGLITZ & J.D. ULLMAN, *Evaluating polynomials at fixed sets of points*, SICOMP 4 (1975) 533-539.

[4] ALDER, A. & V. STRASSEN, *On the algebraic complexity of Associative algebra's*, TCS 15 (1981) 201-212.

[5] ALT, H., *Square rooting is as difficult as multiplication*, Computing 21 (1979) 221-232.

[6] ALT, H., *Functions equivalent to integer multiplication*, ICALP 7 (1980) 30-37, Springer LCS 85.

[7] ALT, H. & J. VAN LEEUWEN, *Complexity of complex division*, FCT 2 (1979) 13-18, Akademie Verlag Berlin.

[8] ATKINSON, M.D., *The complexity of group algebra computations*, TCS 5 (1977) 205-209.

[9] ATKINSON, M.D. & S. LLOYD, *Bounds on the ranks of 3-tensors*, LAA 31 (1980) 19-31.

[10] ATKINSON, M.D. & S. LLOYD, *A special class of 3-tensors*, Manuscript.

[11] ATKINSON, M.D. & N.M. STEPHENS, *On the maximal multiplicative complexity of a family of bilinear forms*, LAA 27 (1978) 1-8.

[12] AUSLANDER, L. & S. WINOGRAD, *Direct sums of bilinear algorithms*, Rep. RC 7916, IBM Yorktown Heights, Oct 1979.

[13] BINI, D., *Relations between EC-algorithms and APA-algorithms, applications*, Nota interna B 7918 (Mar 1979) IEI Pisa.

[14] BINI, D., *Border rank of $p \times q \times r$ tensor and optimal approximation of a pair of bilinear forms*, ICALP 7 (1980) 98-108; Springer LCS 85.

[15] BINI, D., M. CAPOVANI, G. LOTTI & F. ROMANI, *$O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication*, IPL 8 (1979) 66-68.

[16] BINI, D. & M. CAPOVANI, *Lowerbounds of the complexity of linear algebra's*, IPL 9 (1979) 46-47.

[17] BINI, D. & G. LOTTI, *Stability of fast algorithms for matrix multiplication*, Numer. Math. 36 (1980) 63-72.

[18] BINI, D., G. LOTTI & F. ROMANI, *Approximate solutions for the bilinear form computation problem*, SICOMP 9 (1980) 692-697.

[19] BOJANCZYK, A., *Complexity of solving linear systems in different models of computation*, Manuscript Univ. Warsaw. 1981.

[20] BORODIN, A. & S. COOK, *On the number of additions to compute specific polynomials*, SICOMP 5 (1976) 146-157; see also STOC 6 (1974) 342-347.

[21] BORODIN, A. & I. MUNRO, *Evaluating polynomials at many points*, IPL 1 (1972) 66-68.

[22] BORODIN, A. & I. MUNRO, *The computational complexity of algebraic and numerical problems*, Theory of Computation series 1, Amer. Elsevier 1975.

[23] BOURBAKI, N., *Éléments de Mathématique; Algèbre; Chap. II, 3,4; Chap. III, 1*, Herman 1970 (or. Engl. Transl. Addison Wesley 1974).

[24] BRENT, R.P., *Error Analysis of algorithms for matrix multiplication and triangular decomposition using Winograd's identity*, Num. Math. 16 (1970) 145-156.

[25] BRENT, R.P. & H.T. KUNG, *Fast algorithms for manipulating formal power series*, JACM 25 (1978) 581-595.

[26] BRENT, R.P. & J.F. TRAUB, *On the complexity of composition and generalised composition of power series*, SICOMP 9 (1980) 55-66.

[27] BROCKETT, R.W. & D. DOBKIN, *On the optimal evaluation of a set of bilinear forms*, LAA 19 (1978) 207-235; see also STOC 5 (1973) 88-95.

[28] BROCKETT, R.W. & D. DOBKIN, *On the number of multiplications required for matrix multiplication*, SICOMP 5 (1976) 624-628.

[29] BROCKETT, R.W. Oral. Comm. Sep 14, 1976.

[30] BUNCH, J.R. & J.E. HOPCROFT, *Triangular factorisation and inversion by fast matrix multiplication*, Math. Comp. 28 (1974) 231-236.

[31] CHIN, F.Y., *A generalised asymptotic upperbound on fast polynomial evaluation and interpolation*, SICOMP 5 (1976) 682-690.

[32] CHIN, F.Y., *The partial fraction problem and its inverse,* SICOMP 6 (1977) 554-562.

[33] COHEN, J. & M. ROTH, *On the implementation of Strassen's fast multiplication algorithm,* Acta Informatica 6 (1976) 341-355.

[34] COPPERSMITH, D., *Rapid multiplication of rectangular matrices,* SICOMP 11 (1982) 467-471.

[35] COPPERSMITH, D. & S. WINOGRAD, *On the asymptotic complexity of matrix multiplication,* SICOMP 11 (1982) 472-492, see also FOCS 22 (1981).

[36] COURANT, R., *Differential and Integral calculus vol. I, (sec. ed.),* Blackie & Son Ltd. London 1965, p. 361.

[37] DOBKIN, D., *On the complexity of a class of arithmetic computations,* Ph. D. Thesis, Harvard (1973).

[38] DOBKIN, D., *On the optimal evaluation of a set of n-linear forms,* SWAT 14 (1973) 92-102.

[39] DOBKIN, D. & J. VAN LEEUWEN, *The complexity of vector products,* IPL 4 (1976) 149-154.

[40] FATEMAN, R.J., *Polynomial multiplication, powers and asymptotic analysis. Some comments,* SICOMP 3 (1974) 169-213.

[41] FEIG, E., *Certain systems of bilinear forms whose minimal algorithms are all quadratic,* rep RC 8758 (#38298) mar 19 1981 IBM.

[42] FIDUCCIA, C.M., *On obtaining upper bounds on the complexity of matrix multiplication,* in R.E. Miller & J.W. Thatcher (eds) Complexity of computer computations, Plenum NY 1972, 31-40.

[43] FIDUCCIA, C.M., *Fast Matrix multiplication,* STOC 3 (1971) 45-49.

[44] FIDUCCIA, C.M., *Polynomial evaluation via the division algorithm; the fast Fourier transform revisited,* STOC 4 (1972) 88-93.

[45] FIDUCCIA, C.M., *On the algebraic complexity of matrix multiplication,* Ph. D. Thesis, Brown Univ. (1973).

[46] FIDUCCIA, C.M. & Y. ZALCSTEIN, *Algebra's having linear multiplicative complexities,* JACM 24 (1977) 311-331.

[47] FISCHER, M.J., A.R. MEYER & M.S. PATERSON, *Lower bounds on the size of Boolean formulae,* STOC 7 (1975) 37-44.

[48] FISCHER, P.C. & R.L. PROBERT, *Efficient procedures for using matrix algorithms*, ICALP 2 (1974) 413-427; Springer LCS 14.

[49] FISCHER, P.C., *Further schemes for combining matrix algorithms*, ICALP 2 (1974) 428-436; Springer LCS 14.

[50] GANTMACHER, F.R., *Matrizenrechnung Teil II*, VEB Deutscher Verlag der Wissenschaften, Berlin 1959; Satz XII.5.5; p. 53.

[51] GATHEN, J. VON ZUR & V. STRASSEN, *Some polynomials that are hard to compute*, TCS 11 (1980) 331-335.

[52] GASTINEL, N., *Sur le calcul des produits de matrices*, Numer. Math. 17 (1971) 222-229.

[53] GILBERT, J.R., TH. LENGAUER & R.E. TARJAN, *The pebbling problem is complete in polynomial space*, SICOMP 9 (1980) 513-524; see also STOC 11 (1979) 237-248.

[54] GONZALES, F. & J. JA'JA', *On the complexity of computing bilinear forms with (0,1) constants*, J. Comput. Sys. Sci. 20 (1980) 77-95.

[55] GRIGOR'EV, D. YU., *Multiplicative complexity of a pair of bilinear forms and polynomial multiplication*, MFCS 7 (1978) 250-256; Springer LCS 64.

[56] GRIGOR'EV, D. YU., *Some new bounds on tensor rank*, LOMI Preprints E-2-78 Leningrad 1978.

[57] GRIGOR'EV, D.Yu., *Multiplicative complexity of a bilinear form over a commutative ring*, MFCS 10 (1981), 281-286; Springer LCS 118.

[58] GROOTE, H.F. DE, *On the complexity of quaternion multiplication*, IPL 3 (1975) 177-179.

[59] GROOTE, H.F. DE, *On varieties of optimal algorithms for the computation of bilinear mappings, part I; the isotropy group of a bilinear mapping*, TCS 7 (1978) 1-24.

[60] GROOTE, H.F. DE, *On varieties of optimal algorithms for the computation of bilinear mappings, part II; Optimal algorithms for 2 × 2 matrix multiplication*, TCS 7 (1978) 127-148.

[61] GROOTE, H.F. DE, *On varities of optimal algorithms for the computation of bilinear mappings, part III; Optimal algorithms for the computation of XY and YX, where X,Y ∈ $M_2(K)$*, TCS 7 (1979) 239-249.

[62] HARTER, R., *The optimality of Winograd's formula,* Comm. Assoc. Comput. Mach. 15 (1972) 352.

[63] HEINTZ, J. & M. SIEVEKING, *Lower bounds for polynomials with algebraic coefficients,* TCS 11 (1980) 321-330.

[64] HOPCROFT, J.E. & L.R. KERR, *On minimizing the number of multiplications necessary for matrix multiplication,* SIAM J. Appl. Math. 20 (1971) 30-36.

[65] HOPCROFT, J.E. & J. MUSINSKI, *Duality applied to the complexity of matrix multiplication and other bilinear forms,* SICOMP 2 (1973) 159-173; also STOC 5 (1973) 73-87.

[66] HOROWITZ, E., *A sorting algorithm for polynomial multiplication,* JACM 22 (1975) 450-462.

[67] HOWELL, Th.D. & J.C. LAFON, *The complexity of the quaternion product,* rep. TR 75-245 (june 1975) DCS Cornell Univ.

[68] HYAFIL, L., *The power of commutativity,* FOCS 18 (1977) 171-174.

[69] HYAFIL, L. & H.T. KUNG, *The complexity of parallel evaluation of linear recurrences,* JACM 24 (1977) 513-521.

[70] HYAFIL, L., *On the parallel evaluation of multivariable polynomials,* SICOMP 8 (1979) 120-127; also STOC 10 (1978) 193-195.

[71] JA'JA', J., *Optimal evaluation of pairs of bilinear forms,* SICOMP 8 (1979) 433-462; also STOC 10 (1978) 173-183.

[72] JA'JA', J., *On the complexity of bilinear forms with commutativity,* SICOMP 9 (1980) 713-728; also STOC 11 (1979) 197-208.

[73] JA'JA', J., *Time space trade offs for some algebraic problems,* STOC 12 (1980) 339-350.

[74] JA'JA', J., *Computation of bilinear forms over finite fields,* JACM 27 (1980) 822-830.

[75] KEDEM, Z.M., *Combining dimensionality and rate of growth arguments for establishing lower bounds on the number of multiplications and divisions,* JACM 26 (1979) 582-601; also STOC 6 (1974) 334-341.

[76] KIRKPATRICK, D., *On the additions necessary to compute certain functions,* STOC 4 (1972) 94-101.

[77] KIRKPATRICK, D., & Z.M. KEDEM, *Addition requirements for rational functions*, SICOMP 6 (1977) 188-199.

[78] KLETTE, R., *Fast matrix multiplication by Boolean RAM in linear storage*, MFCS 7 (1978) 308-314; Springer LCS 64.

[79] KNUTH, D.E., *The art of computer programming, Vol. 2 (seminumerical algorithms)*, Addison Wesley 1969.

[80] KNUTH, D.E., *The art of Computer programming, vol 2* (revised edition) Exc. 4.6.4 60-64 pp 504-505 + solutions 654-655, Addison Wesley 1981.

[81] KREZMAR, A., *On memory requirements of Strassen's Algorithms*, MFCS 5 (1976) 404-407; Springer LCS 45.

[82] KRONECKER, L. *Mathematische Werke Vol. I*, ed. K. Hensel; Chelsey Publ. Co. 1968.

[83] KRUSKAL, J.B., *Three way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, LAA 18 (1977) 95-138.

[84] KUCK, D. & Y. MURAOHA, *Bounds on the parallel evaluation of arithmetic expressions using associativity and commutativity*, Acta Informatica 3 (1974) 203-216.

[85] KUNG, H.T., *On computing reciprocals of power series*, Num. Math. 22 (1974) 341-348.

[86] KUNG, H.T. & D.M. TONG, *Fast algorithms for partial function decomposition*, SICOMP 6 (1977) 582-593.

[87] KUNG, H.T. & J.F. TRAUB, *All algebraic functions can be computed fast*, JACM 25 (1978) 245-260.

[88] LADERMAN, J.D., *A non commutative algorithm for multiplying 3 × 3 matrices using 23 multiplications*, Bull. AMS 82 (1976) 126-128.

[89] LADNER, R.E. & M.J. FISCHER, *Parallel prefix computation*, JACM 27 (1980) 831-838.

[90] LAFON, J.C., *Optimum computation of p-linear forms*, LAA 10 (1975) 225-240.

[91] LAFON, J.C., *Sur le produit de deux quaternions*, C.R. Acad. Sci. Paris 280 (1975), Ser 1, 665-668.

[92] LAFON, J.C., *Base tensorielle des matrices de Hankel (ou de Touplitz), applications,* Num. Math. 23 (1975) 349-361.

[93] LAFON, J.C., *Evaluation simultane de plusieurs formes bilinéaires,* in Journées Algorithmiques, Astérisque 38-39 (1976) 117-130.

[94] LAMAGNA, E.A. & J.E. SAVAGE, *Combinatorial complexity of some monotone functions,* SWAT 15 (1974) 140-144.

[95] LASKOWSKI, SH. J. & D.P. DOBKIN, *The structure and rank of $m \times p \times q$ tensors; the heuristic approach, rep.* DCS Yale 138, apr. 1978.

[96] LASKOWSKI, SH. J., *Computing lower bounds on tensor rank over finite fields,* rep. CS 80-18 Penn. State Univ., Jul. 1980.

[97] LAUTEMAN, C., *On the tensor rank of matrices,* rep. 79-18 TU. Berlin, Fachber. Informatik 20, sep. 1979.

[98] LAZARD, D., *Algorithms fundamentaux en algèbre commutative,* in Journées Algorithmiques, Astérisque 38-39 (1976) 131-138.

[99] LAZARD, D., *Sur le produit de matrices,* Gazette des mathématiciens 15 (1980) 27-48.

[100] MCCOLL, W.F. & M.S. PATERSON, *The depth of all Boolean functions,* SICOMP 6 (1977) 373-380.

[101] MCKAY, J.H., *The William Lowel Putnam mathematical competition, Problem A-6,* Amer. Math. Monthly 80 (1973) 170-179.

[102] MANDL, R. & T. VARI, *Computational complexity of inner products of vectors (and that of other bilinear forms) over a non commutative ring (auxiliary functions allowed),* SICOMP 4 (1975) 49-55.

[103] MEHLHORN, K. & Z. GALIL, *Monotone switching circuits and Boolean matrix products,* Computing 16 (1976) 99-111.

[104] MILLER, R.E. & J.W. THATCHER (ed.), *Complexity of computer algorithms; proc. Complexity of computer computation workshop, IBM Yorktown Heights Mar 20-22 1972.* Plenum NY 1972.

[105] MILLER, W., *Computational complexity and numerical stability,* SICOMP 4 (1975) 97-107; also STOC 6 (1974) 317-322.

[106] MOENCK, R.T., *Another polynomial homomorfism,* Acta Informatica 6 (1976) 153-169.

[107] MORGENSTERN, J., *Note on a lower bound of the linear complexity of the fast Fourier transform,* JACM 20 (1973) 305-306.

[108] MORGENSTERN, J., *The linear complexity of computation,* JACM 22 (1975) 184-194.

[109] MORGENSTERN, J., *Transformations de Fourier discrètes,* in Journées Algorithmiques, Astériques 38-39 (1976) 159-168.

[110] MORGERA, S.D., *Efficient synthesis and implementation of large discrete Fourier transformations,* SICOMP 9 (1980) 251-272.

[111] MUNRO, I., *Some results concerning efficient and optimal algorithms,* STOC 3 (1971) 40-44.

[112] MUNRO, I., *Problems related to matrix multiplication,* in R. Rustin (ed.) *Computational Complexity,* Algorithmic Press Inc. NY 1973, pp. 137-152.

[113] PAN, V. YA., *Methods of computing values of polynomials,* Russian Math. Surv. 21 (1966) 105-136 (Russian).

[114] PAN, V. YA., *On schemes for the computation of products and inverses of Matrices,* Russian Math. Surv. 27 (1972) 249-250 (Russian).

[115] PAN, V. YA., *Computational complexity of computing polynomials over the fields $\mathbb{C}$ and $\mathbb{R}$,* rep. IBM RC 7754, may 1979; also STOC 10 (1978) 162-172.

[116] PAN, V. YA., *Strassen's algorithm is not optimal, trilinear technique of aggregating, uniting and canceling for construction of fast matrix operations,* FOCS 19 (1978) 166-176.

[117] PAN, V. YA., *New combinations of methods for the acceleration of Matrix multiplication,* rep. TR 80-1, SUNY Albany.

[118] PAN, V. YA., *New fast algorithms for matrix operations,* SICOMP 9 (1980) 321-342, also FOCS 20 (1979) 28-38.

[119] PAN, V. YA., *Convolution of vectors over the real field of constants by evaluation - interpolation algorithms,* J. Algorithms 1 (1980) 297-300.

[120] PAN, V. YA., $O(n^{2.52})$ *binary bit-operations for approximate evaluation of the product of $n \times n$ matrices (multiprecision arithmetic for stabilization of computations)* rep. TR 80-2, SUNY Albany.

[121] PAN, V. YA., *The bit-operation complexity of the convolution of vectors and of the DFT.* rep. TR 80-6, SUNY Albany.

[122] PAN, V. YA., *An approach to the analysis of bilinear algorithms,* rep. TR 80-11, SUNY Albany.

[123] PATERSON, M.S. & L.J. STOCKMEYER, *On the number of non-scalar multiplications necessary to evaluate polynomials,* SICOMP 2 (1973) 60-66.

[124] PATERSON, M.S., *Complexity of matrix algorithms,* in J.W. de Bakker, (ed.), *Foundations of Computer Science,* Math. Centre Tracts 63, Amsterdam 1975, pp. 181-215.

[125] PAUL, W.J., *A 2.5n lower bound on the combinational complexity of Boolean functions,* SICOMP 6 (1977) 427-443; also STOC 7 (1975) 27-36.

[126] PIPPENGER, N., *The realisation of monotone Boolean functions,* STOC 8 (1976) 204-210.

[127] PIPPENGER, N., *Information theory and the complexity of Boolean functions,* Math. Systems Theory 10 (1977) 129-167; also FOCS 16 (1975) 113-118.

[128] PIPPENGER, N., *The complexity of monotone Boolean functions,* Math. Systems Theory 11 (1978) 289-316.

[129] PIPPENGER, N., *Computational complexity in algebraic function fields,* FOCS 20 (1979) 61-63.

[130] PIPPENGER, N., *On the evaluation of powers and monomials,* SICOMP 9 (1980) 230-250.

[131] PIPPENGER, N., *Computational complexity of algebraic functions,* rep. IBM RC 8113, feb 1980.

[132] PRATT, V.R., *The power of negative thinking in multiplying Boolean matrices,* STOC 6 (1974) 80-83,

[133] PRATT, V.R., *The effect of basis on size of Boolean expressions,* FOCS 16 (1975) 119-121.

[134] PREPARATA, F.P. & J.E. VUILLEMIN, *Area-time optimal VLSI networks for multiplying matrices,* IPL 11 (1980) 77-80.

[135] PROBERT, R.L., *On the additive complexity of matrix multiplication,* SICOMP 5 (1976) 187-203.

[136] PROBERT, R.L., *An extension of computational duality to sequences of bilinear computations,* SICOMP 7 (1978) 91-98.

[137] REHAV, L., *On the number of multiplications/divisions evaluating a polynomial with auxiliary functions,* SICOMP 4 (1975) 381-392.

[138] REINGOLD, E.M. & A.I. STOCKS, *Simple proofs of lower bounds for polynomial evaluation,* in R.E. Miller & J.W. Thatcher (eds), *Complexity of computer computations,* Plenum NY 1972, pp. 21-30.

[139] ROMANI, F., *Shortest-path problem is not harder than matrix multiplication,* IPL 11 (1980) 134-136.

[140] RUSTIN, R., *Computational complexity, Courant computer science symposium 7, Oct. 25-26 1971,* Algorithmics Press Inc. NY 1973.

[141] SANKY, L.C., *Fast parallel matrix inversion algorithm,* SICOMP 5 (1976) 618-623.

[142] SAVAGE, J.E., *An algorithm for the computation of linear forms,* SICOMP 3 (1974) 150-158.

[143] SAVAGE, J.E., *The complexity of computing,* Wiley 1976.

[144] SAVAGE, J.E., *Area-time tradeoffs for matrix multiplication and related problems in VLSI models,* J. Comput. Systems Sci. 22 (1981) 230-242.

[145] SCHACHTEL, A., *A non commutative algorithm for multiplying 5 × 5 matrices using 103 multiplications,* IPL 7 (1978) 180-182.

[146] SCHÖNHAGE, A., *Multiplikation grosser Zahlen,* Computing 1 (1966) 182-196.

[147] SCHÖNHAGE, A. & V. STRASSEN, *Schnelle Multiplikation grosser Zahlen,* Computing 7 (1971) 281-292.

[148] SCHÖNHAGE, A., *Unitäre Transformationen grosser Matrizes,* Num. Math. 20 (1973) 409-417.

[149] SCHÖNHAGE, A., *Schnelle Berechnung von Kettenbruchentwicklungen,* Acta Informatica 1 (1974) 139-144.

[150] SCHÖNHAGE, A., *An elementary proof for Strassen's degree bound,* TCS 3 (1976) 267-272.

[151] SCHÖNAGE, A., *Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2*, Acta Informatica 7 (1977) 395-398.

[152] SCHÖNHAGE, A., *Partial and total matrix multiplication*, Manuscript, Univ. Tübingen June 1979. Zie ook: SICOMP 10 (1981) 434-455.

[153] SCHNORR, C.P., *Zwei lineäre untere Schranken für die Komplexität Boolsche Funktionen*, Computing 13 (1974) 155-171.

[154] SCHNORR, C.P., *A lower bound on the number of additions in monotone computions*, TCS 2 (1976) 305-315.

[155] SCHNORR, C.P., *Improved lower bounds on the number of multiplications/ divisions which are necessary to evaluate polynomials*, TCS 7 (1978) 251-261; also MFCS 6 (1977) 135-147, Springer LCS 53.

[156] SCHNORR, C.P., *A 3n-lower bound on the network complexity of Boolean functions*, TCS 10 (1980) 83-92.

[157] SCHNORR, C.P., *An extension of Strassen's degree bound*, SICOMP 10 (1981) 371-382.

[158] SCHNORR, C.P., *How many polynomials can be approximated faster than they can be evaluated*, IPL 12 (1981) 76-78.

[159] SCHNORR, C.P. & J.P. VAN DE WIELE, *On the additive complexity of polynomials*, TCS 10 (1980) 1-18.

[160] SCHNORR, C.P., *Mathematisches Forschungsinstitut Oberwolfach, Tagungsbericht 44/1979; Complexity theory; October 21-28 1979*, EATCS Bulletin 10 (1980) 102-125.

[161] SCHWARZ, H.R., *The fast Fourier transform for general Order*, Computing 19 (1978) 341-350.

[162] SHAMOS, M.I. & G. YUVAL, *Lower bounds from complex function theory*, FOCS 17 (1976) 268-273.

[163] SHAW, M. & J.F. TRAUB, *On the number of multiplications for the evaluation of a polynomial and some of its derivatives*, JACM 21 (1974) 161-168.

[164] SIEVEKING, M., *An algorithm for division of power series*, Computing 10 (1972) 153-156.

[165] SPIESS, J., *Untersuchungen des Zeitgewinns durch neue Algorithmen zur Matrix Multiplikation*, Computing 17 (1976) 23-26.

[166] STOCKMEYER, L.J., *On the combinational complexity of certain symmetric Boolean functions*, Math.Systems Theory 10 (1977) 323-336.

[167] STRASSEN, V., *Gaussian elimination is not optimal*, Num. Math. 13 (1969) 354-356.

[168] STRASSEN, V., *Evaluation of rational functions*, in R.E. Miller & J.W. Thatcher (eds) *Complexity of computer computations*, Plenum NY 1972, pp. 1-10.

[169] STRASSEN, V., *Berechnung und Program I*, Acta Informatica 1 (1972) 320-335.

[170] STRASSEN, V., *Berechnung und Program II*, Acta Informatica 2 (1973) 64-79.

[171] STRASSEN, V., *Berechnungen in partiellen Algebren endlichen Typs*, Computing 11 (1973) 181-196.

[172] STRASSEN, V., *Vermeidung von Divisionen*, Crelle J. Reine und Angew. Math. 264 (1973) 184-202.

[173] STRASSEN, V., *Die Berechnungskomplexität von elementar symmetrischen Funktionen und interpolations Koeffizienten*, Num. Math. 20 (1973) 238-251.

[174] STRASSEN, V., *Polynomials with rational coefficients which are hard to compute*, SICOMP 3 (1974) 128-149.

[175] STRASSEN, V., *Computational complexity over finite fields*, SICOMP 5 (1976) 324-332.

[176] STRASSEN, V., *Einige Resultate über Berechnungskomplexität*, Jahrber. Dt. Math. Verein 78 (1976) 1-8.

[177] SYKORA, D., *A fast non-commutative algorithm for matrix multiplication*, MFCS 6 (1977) 504-512; Springer LCS 53.

[178] TOMPA, M., *Time-space tradeoffs for computing functions using connectivity properties of their circuits*. J. Comput. Systems Sci. 20 (1980) 118-132.

[179] TSAO, N., *The numerical instability of Bini's algorithm*. IPL 12 (1981) 17-19.

[180] VALIANT, L.G., *On non-linear lower bounds in computational theory*, STOC 7 (1975) 45-53.

[181] VALIANT, L.G., *General context-free recognition in less than cubic time*, J. Comput. Systems Sci. 10 (1975) 308-315.

[182] VALIANT, L.G., *Negation can be exponentially powerfull*, TCS 12 (1980) 303-314; also STOC 11 (1979) 189-196.

[183] VAN LEEUWEN, J. & P. VAN EMDE BOAS, *Some elementary proofs of lower bounds in complexity theory*, LAA 19 (1978) 63-80.

[184] VAN LEEUWEN, J., *Über Programmeffizienz und algebraischen Komplexität*. Informatik Spectrum 3 (1980) 172-180.

[185] WIELE, J.P. VAN DE, *An optimal lower bound on the number of total* LAA *operations to compute 0-1 polynomials over the field of complex numbers*, FOCS 19 (1978) 159-165.

[186] WINOGRAD, S., *A new algorithm for inner product*, IEEE Trans. C-17 (1968) 693-694.

[187] WINOGRAD, S., *On the number of multiplications necessary to compute certain functions*. Comm. Pure Appl. Math. 23 (1970) 165-179.

[188] WINOGRAD, S., *On the algebraic complexity of the inner product*, LAA 4 (1971) 377-379.

[189] WINOGRAD, S., *On the multiplication of 2 × 2 matrices*, LAA 4 (1971) 381-388.

[190] WINOGRAD, S., *On the parallel evaluation of certain arithmetic expressions*, JACM 22 (1975) 477-492.

[191] WINOGRAD, S., *The effect of the field of constants on the number of multiplications*, FOCS 16 (1975) 1-2.

[192] WINOGRAD, S., *On computing the discrete Fourier transform*, Proc. Acad. Nat. Sci. USA 73 (1976) 1005-1006.

[193] WINOGRAD, S., *Some bilinear forms whose multiplicative complexity depends on the field of constants*, Math. Systems Theory 10 (1977) 169-180.

[194] WINOGRAD, S., *On the multiplicative complexity of the discrete Fourier transform*, rep. IBM RC 7373, Oct 1978.

[195] WINOGRAD, S., *On multiplication in algebraic extension fields*, TCS 8 (1979) 359-377.

[196] WINOGRAD, S., *Arithmetic complexity of computations*, CBNS-NSF regional conference series in appl. math. 33 (1980).

[197] WINOGRAD, S., *On multiplication of polynomials modulo a polynomial*, SICOMP 9 (1980) 225-229.

[198] WONG, C.K. & R.J. LIPTON, *Addition chain methods for the evaluation of specific polynomials*, SICOMP 9 (1980) 121-125.

[199] WOZNIAKOWSKI, H., Oral. Comm. Dec 04 1980.

[200] YAO, A.C., *On the evaluation of powers*, SICOMP 5 (1976) 100-103.

[201] YUVAL, G., *An algorithm for finding all the shortest paths using* $n^{2.81}$ *infinite precision multiplications*, IPL 4 (1976) 155-156.

[202] YUVAL, G., *A simple proof of Strassen's Result*, IPL 7 (1978) 285-286.

PRIMALITEIT EN FACTORIZATIE

H.W. LENSTRA, Jr.

1. AANVANG

Zij $n > 1$ een geheel getal. We beschouwen de volgende twee problemen:

(a) (*primaliteit*) is n priem?

(b) (*factorizatie*) zo nee, vind $a,b > 1$ met $n = ab$.

We zullen niet een volledig overzicht geven van alle voorgestelde methoden, maar ons beperken tot de *theoretisch* best bekende. Voor al het andere, zoals *practische* methoden, geschiedkundige opmerkingen, toepassingen en verdere literatuurverwijzingen zie men WILLIAMS [15], GUY [5], MONIER [9], SCHNORR [12], KNUTH [6] en voorts de uitgewerkte teksten van voordrachten, gehouden tijdens de studieweek "Getaltheorie en Computers" [4].

Zoals gebruikelijk zullen we een algoritme *goed* noemen als de rekentijd begrensd wordt door een polynoom in de lengte van de input. Voor problemen (a) en (b) is de input het getal n, dat binaire lengte $[\log n/\log 2] + 1$ heeft. De lengte van de input heeft dus dezelfde orde van grootte als $\log n$.

De bekendste methode om de problemen (a) en (b) op te lossen bestaat uit het achtereenvolgens proberen of n deelbaar is door $2,3,4,\ldots,[\sqrt{n}]$. Dit kan $\sqrt{n}$ stappen kosten, hetgeen exponentieel is in de lengte van de input. Deze algoritme is dus niet "goed".

Voordat men op zoek gaat naar een kort bewijs dat n priem is, of een kort bewijs dat n samengesteld is, kan men zich afvragen of een dergelijk bewijs wel bestaat. In deze richting hebben we ten eerste het volgende resultaat; onder een *rekenkundige bewerking* verstaan we een optelling, aftrekking en vermenigvuldiging van twee gehele getallen.

STELLING 1. *Als* n *samengesteld is bestaat hiervoor een bewijs bestaande uit* O(1) *rekenkundige bewerkingen. Als* n *priem is, dito.*

BEWIJS. Voor samengestelde n is de stelling triviaal: om te bewijzen dat n samengesteld is is het voldoende twee gehele getallen a,b > 1 op te schrijven en de enkele vermenigvuldiging uit te voeren waaruit blijkt dat ab = n.

Voor n priem maken we gebruik van een nevenresultaat van de negatieve oplossing van het tiende probleem van Hilbert. Dit resultaat zegt dat er een polynoom

$$f \in \mathbb{Z}[\underline{A},\underline{B},\ldots,\underline{Z}]$$

in 26 variabelen bestaat met de eigenschap dat de verzameling priemgetallen samenvalt met de verzameling *positieve* waarden die f aanneemt als niet-negatieve gehele getallen voor $\underline{A},\underline{B},\ldots,\underline{Z}$ gesubstitueerd worden. Om te bewijzen dat n priem is is het dus voldoende 26 gehele getallen A,B,...,Z op te schrijven en de begrensde hoeveelheid rekenwerk uit te voeren waaruit blijkt dat n = f(A,B,...,Z). Men vindt dat niet meer dan 87 rekenkundige bewerkingen hiervoor vereist zijn. Dit bewijst Stelling 1. □

Het belang van Stelling 1 is om twee redenen louter theoretisch. In de eerste plaats vanwege het non-deterministische karakter van de methode: het bestaan van zekere bewijzen wordt verzekerd, maar er wordt niet bij verteld hoe men ze snel vindt. In de tweede plaats is het meten van de lengte van een bewijs door middel van het aantal rekenkundige bewerkingen volslagen onrealistisch. Wil men met het in bovenstaand bewijs bedoelde polynoom f bewijzen dat n priem is, dan zal tenminste één van A,B,...,Z groter zijn dan

$$n^{n^{n^{n}}}.$$

Het is kennelijk onrealistisch om een rekenkundige bewerking met getallen van deze orde van grootte als één stap te tellen. Daarom zullen we in het vervolg - behalve in Stelling 6 - *bit-operaties* tellen; deze kunnen we definiëren als rekenkundige bewerkingen op getallen van één (binair) cijfer.

Een vermenigvuldiging van twee getallen < n vereist niet meer dan $O((\log n)^2)$ bit-operaties, dus er geldt de volgende stelling:

STELLING 2. *Als* n *samengesteld is bestaat hiervoor een bewijs bestaande uit* $O((\log n)^2)$ *bit-operaties.*

Met behulp van snellere vermenigvuldig-routines kan men dit resultaat verbeteren tot $O((\log n)^{1+\varepsilon})$, voor elke $\varepsilon > 0$. Evenzo kan in de volgende stelling de exponent 4 door $3+\varepsilon$ vervangen worden, voor elke $\varepsilon > 0$.

STELLING 3 (PRATT [11]). *Als* n *priem is bestaat hiervoor een bewijs bestaande uit* $O((\log n)^4)$ *bit-operaties.*

BEWIJS. We nemen n oneven. Uit elementaire getaltheorie volgt dat n priem is dan en slechts dan als er een geheel getal a is, $0 < a < n$, waarvoor geldt

$$a^{(n-1)/2} \equiv -1 \bmod n,$$
$$a^{(n-1)/q} \not\equiv 1 \bmod n \text{ voor elke priemfactor } q \text{ van } n-1.$$

Dus, om te bewijzen dat n priem is schrijven we een geheel getal a op, $0 < a < n$, we schrijven de ontbinding van $n-1$ op:

$$(1) \qquad n-1 = q_0 q_1 \ldots q_t \qquad \text{met } q_0 = 2,$$

we verifiëren dat

$$(2) \qquad a^{(n-1)/2} \equiv -1 \bmod n,$$

$$(3) \qquad a^{(n-1)/q_i} \not\equiv 1 \bmod n \qquad \text{voor } 1 \le i \le t$$

en we controleren recursief:

$$(4) \qquad q_i \text{ is priem} \qquad (1 \le i \le t).$$

Dit alles vereist t vermenigvuldigingen in (1), en t+1 machtsverheffingen in (2) & (3), plus wat voor (4) vereist is. Geven we met f(n) het totale aantal vermenigvuldigingen en machtsverheffingen aan, dan geldt dus

$$f(n) \le t + t + 1 + \sum_{i=1}^{t} f(q_i);$$

hier definiëren we $f(2) = 1$. Inductief bewijzen we dat $f(n) \le 3(\log n/\log 2) - 2$. Dit geldt voor $n = 2$, en als het voor de q_i geldt, dan

$$f(n) \le 2t + 1 + \sum_{i=1}^{t} (3(\log q_i/\log 2) - 2) = \left(\sum_{i=0}^{t} 3(\log q_i/\log 2)\right) - 2 =$$

$$= 3(\log(n-1)/\log 2) - 2 < 3(\log n/\log 2) - 2,$$

zoals verlangd.

Er zijn dus niet meer dan O(log n) vermenigvuldigingen en machtsverheffingen nodig. Elke machtsverheffing in (2), (3) kan men uitvoeren door middel van O(log n) kwadrateringen en vermenigvuldigingen modulo n. Elke vermenigvuldiging, kwadratering of vermenigvuldiging modulo n (of een getal kleiner dan n) kost $O((\log n)^2)$ bit-operaties. Dit leidt tot de grens $O((\log n)^4)$, waarmee Stelling 3 bewezen is. □

Stellingen 2 en 3 hebben nog het eerste gebrek van Stelling 1: het non-deterministische karakter van de methode. Desondanks is het gegeven bewijs van Stelling 3 niet uitsluitend van theoretische waarde: er zijn verscheidene in de practijk toegepaste primaliteitstests waarvan de grondgedachte dezelfde is. De voornaamste moeilijkheid is dan het vinden van de priemfactorontbinding van n-1. Er zijn varianten waarvoor een gedeeltelijke ontbinding van n-1 ook al voldoende is. Lukt het niet om genoeg factoren van n-1 te vinden, dan zijn er verwante tests waarvoor men factoren van n+1 dient te kennen, en beide soorten tests kunnen ook gecombineerd worden. Het vervolg van n-1, n+1 luidt niet n-2, n+2, n-3, ... zoals men zou kunnen denken, maar

$$n^2+n+1,\ n^2+1,\ n^4+n^3+n^2+n+1,\ n^2-n+1,\ \ldots$$

waarbij de k-de term gegeven is door

$$\Phi_k(n) = \prod_{1\le a\le k,\, ggd(a,k)=1} (n - e^{2\pi i a/k}),$$

(dus $\Phi_1(n) = n-1$, $\Phi_2(n) = n+1$).

Voor getallen n van de vorm $n = 2^k \pm 1$ is $n \mp 1$ eenvoudig in factoren te ontbinden, en in deze gevallen blijken de bovengenoemde tests een bijzonder eenvoudige vorm aan te nemen. Dat is een plezierige samenloop van omstandigheden, want priemgetallen van de vorm $2^k \pm 1$ spelen een bijzondere rol in de wiskunde: met behulp van de priemgetallen van de vorm 2^k+1 (*Fermat*-priemgetallen) wist *Gauss* alle getallen n te karakteriseren waarvoor een regelmatige n-hoek met passer en liniaal te construeren is; en priemgetallen van de vorm 2^k-1 (*Mersenne*-priemgetallen) komen voor in een stelling van *Euclides* en *Euler* over volmaakte getallen. De enige bekende Fermat-priemgetallen (met $k > 0$) zijn 3,5,17,257 en 65537, en er zijn redenen om aan te nemen dat dit ze alle zijn. Daarentegen vermoedt men dat er oneindig veel

Mersenne-priemgetallen zijn. Met deze stand van zaken is het weinig verbazend dat het "grootst bekende priemgetal", zoals dat tot ons komt via de nieuwsmedia en het *Guinness book of records*, steeds een Mersenne-priemgetal is; op het ogenblik is $2^{44497}-1$ de gelukkige.

Voordat we deze het theoretische kader van deze voordracht ietwat te buiten gaande uitweiding beeindigen noemen we nog een tweede aardige toepassing van de primaliteitstests die op ontbindingen van n±1 berusten, namelijk de jacht op *priemgetaltweelingen*. Een priemgetaltweeling is een tweetal priemgetallen met verschil 2, zoals 3, 5 of 101, 103. Men vermoedt dat er oneindig veel priemgetaltweelingen bestaan; de grootst bekende is

$$256200945 \cdot 2^{3426} \pm 1 = 2^{3426} \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 113 \cdot 151 \pm 1.$$

Zoals men ziet is het getal dat tussen de tweeling-priemgetallen in ligt uit kleine factoren opgebouwd, zodat de ontbinding ervan volledig bekend is. Dit maakt het mogelijk de primaliteit van beide getallen te bewijzen door op de grootste een n-1 - test toe te passen, en op de kleinste een n+1 - test.

Voor meer details over deze zaken verwijzen we naar de aan het begin aangehaalde literatuur.

We keren terug tot onze theoretische beschouwingen. Stellingen 2 en 3 impliceren dat, in modern jargon, het probleem of een getal priem is behoort to NP ∩ coNP. Behoort het, zo zal de expert zich dan direct afvragen, wellicht ook tot de klasse P? Dat wil zeggen, is er "goede" en *deterministische* algoritme die beslist of een getal n priem is? Het antwoord hierop luidt waarschijnlijk bevestigend, maar zoals de zaken nu staan moeten we voor een definitief antwoord wachten op het bewijs van de zgn. gegeneraliseerde Riemann-hypothese. Deze hypothese zegt iets over de ligging van de nulpunten van zekere in de analytische getaltheorie optredende complexe functies, en een bewijs ervan zou verreikende consequenties hebben. Voor een precieze formulering verwijzen we naar de literatuur.

STELLING 4 (G.L. MILLER [8]). *Laat de gegeneraliseerde Riemann-hypothese waar zijn. Dan is er een algoritme die in* $O((\log n)^5)$ *stappen beslist of* n *priem is.*

Beneden geven we een beschrijving van de algoritme. Zolang de Riemann-hypothese onbewezen blijft is het interessant na te gaan welke rol de hypothese in de algoritme speelt. Het blijkt dat de terminatie van de

algoritme in $O((\log n)^5)$ stappen niet van de hypothese afhankelijk is, maar de correctheid van het antwoord wel. Nauwkeuriger geformuleerd: als de algoritme het getal n samengesteld verklaart is n inderdaad samengesteld; maar als de algoritme n priem verklaart, kan het zijn dat n samengesteld is, en de Riemann-hypothese fout.

Neem aan dat n oneven is, en schrijf $n-1 = u \cdot 2^t$, met u oneven en $t \geq 1$. In navolging van Rabin noemen we een geheel getal a een *getuige* voor de samengesteldheid van n, of kortweg een getuige voor n, als de volgende drie voorwaarden vervuld zijn:

(5) $\quad a \not\equiv 0 \bmod n,$

(6) $\quad a^u \not\equiv 1 \bmod n,$

(7) $\quad a^{u \cdot 2^i} \not\equiv -1 \bmod n \quad$ voor $i = 0,1,\ldots,t-1.$

Of a al dan niet een getuige van n is hangt alleen van a modulo n af; dus we mogen ons beperken tot $0 \leq a < n$. Voor zo'n a kan men in $O((\log n)^3)$ stappen nagaan of (5), (6) en (7) vervuld zijn.

Getuigen zijn betrouwbaar: als a een getuige is voor de samengesteldheid van n dan is n inderdaad samengesteld. Neem namelijk aan dat n priem is. Uit (5) en de stelling van Fermat (zie beneden) volgt dan dat

$$a^{n-1} \equiv 1 \bmod n,$$

dus van de rij

$$a^u,\ a^{u \cdot 2},\ \ldots,\ a^{u \cdot 2^t}$$

is de laatste term 1 mod n. De eerste niet, wegens (6). Zij $b = a^{u \cdot 2^i}$ de laatste term in de rij die niet 1 mod n is. Dan geldt $0 \leq i \leq t-1$, en $b^2 = a^{u \cdot 2^{i+1}} \equiv 1 \bmod n$. Dus n deelt $b^2-1 = (b+1)(b-1)$, maar uit $b \not\equiv 1 \bmod n$ volgt dat n geen deler is van b-1. Omdat n priem is impliceert dit dat n een deler van b+1 is, dus $b \equiv -1 \bmod n$, in tegenspraak met (7).

In de practijk zijn getuigen niet lastig te vinden, als n tenminste samengesteld is: Rabin heeft bewezen dat in dat geval tenminste 75% van de getallen $1,2,\ldots,n-1$ uit getuigen voor n bestaat. Dit leidt tot een probabilistische primaliteitstest: trek honderd willekeurige getallen uit $\{1,2,\ldots,n-1\}$; is er een getuige voor n bij, dan is n samengesteld; en is er geen getuige bij dan verklaart men n priem. In ten hoogste één op de 4^{100} gevallen leidt dit tot een verkeerd antwoord. Dit is voor de commerciële productie van priemgetallen een aanvaardbaar risico, maar het levert

geen bewijs van Stelling 4. Om dit te doen vervangen we de honderd willekeurige getallen door de getallen a met $1 < a < 70 \cdot (\log n)^2$; uit de Riemann-hypothese kan men namelijk afleiden dat elke samengestelde n een getuige in dit interval heeft. Vindt men dus geen getuige, dan moet n priem zijn, of de Riemann-hypothese fout. Dit besluit onze schets van het bewijs van Stelling 4. □

Zonder Riemann-hypothese blijven we zitten met een slechte algoritme:

STELLING 5 (ADLEMAN, POMERANCE, RUMELY [1,2]). *Er is een algoritme die in* $O((\log n)^{c \log\log\log n})$ *stappen beslist of* n *priem is, voor* $n > e^e$. *Hier geeft* c *een effectief berekenbare constante aan.*

Het bewijs van Stelling 5 vereist gecompliceerde beschouwingen uit zowel de algebraische als de analytische getaltheorie, die buiten het kader van deze voordracht vallen. Met een verbeterde versie van de algoritme uit Stelling 5 (zie LENSTRA [7]) verwacht men in de practijk de primaliteit van getallen van ruim honderd cijfers gemakkelijk te kunnen beslissen.

De meeste van de bovenbeschreven primaliteitstests vertonen een verbazend gedrag wanneer men ze op niet-priemgetallen n toepast: men krijgt te horen dat n samengesteld is, maar een factor van n wordt er niet bijgeleverd, en is uit de gedane berekeningen ook niet rechtstreeks af te leiden. Stel bijvoorbeeld dat we weten dat n samengesteld is omdat we een geheel getal a hebben gevonden met

$$a^{n-1} \not\equiv 1 \bmod n, \quad \gcd(a,n) = 1,$$

hetgeen volgens de stelling van Fermat voor n priem onmogelijk is. Om in te zien waarom dit ons geen factor van n levert, moeten we nagaan hoe de stelling van Fermat bewezen wordt. Dit kan men doen door op te merken dat de afbeelding $i \mapsto ai \pmod n$ een permutatie van $\{1,2,\ldots,n-1\}$ is, en dat bijgevolg

$$a^{n-1} \cdot (n-1)! = \prod_{i=1}^{n-1} (ai) \equiv \prod_{i=1}^{n-1} i = (n-1)! \bmod n.$$

Als nu $a^{n-1} \not\equiv 1 \bmod n$ dan moet $(n-1)!$ een factor gemeenschappelijk hebben met n, hetgeen ons niets meer of minder vertelt dan dat n samengesteld is.

Ondertussen is het de moeite waard op te merken dat snelle manieren om faculteiten of binomiaalcoëfficiënten modulo n te berekenen behulpzaam kunnen zijn bij factorizatie. Deze opmerking ligt ten grondslag aan de volgende

twee stellingen, in de eerste waarvan weer, als in Stelling 1, op weinig realistische wijze het aantal rekenkundige bewerkingen geteld wordt; bovendien beschouwen we een deling met rest nu ook als een rekenkundige bewerking.

STELLING 6 (SHAMIR [13]). *Er bestaat een algoritme die voor elke samengestelde* n *een niet-triviale deler van* n *levert, en die niet meer dan* $O(\log n)$ *rekenkundige bewerkingen vereist.*

BEWIJS. Het getal n is samengesteld dan en slechts dan als $1 < \text{ggd}(a_0!,n) < n$ voor een positief geheel getal a_0. Aangezien ggd(a!,n) als functie van a monotoon niet-dalend is, en in a = 1, n de waarden 1,n aanneemt, kan een dergelijke a_0 door middel van $O(\log n)$ bisecties bepaald worden, mits we weten hoe ggd(a!,n) te berekenen.

Als we a! kennen, kunnen we de ggd met de Euclidische algoritme in $O(\log n)$ rekenkundige stappen bepalen. Ter berekening van a! passen we de formules

$$(2b+1)! = (2b+1)\cdot(2b)!,$$
$$(2b)! = (b!)^2\cdot\binom{2b}{b}$$

$O(\log a)$ keer toe. De benodigde binomiaalcoëfficiënt $\binom{2b}{b}$ bepaalt men door naar het middelste blok van n binaire cijfers in $(2^n+1)^{2b} = \sum_{i=0}^{2b}\binom{2b}{i}\cdot 2^{in}$ te kijken, voor $2b < n$, waarbij de machtsverheffing met $O(\log(2b))$ vermenigvuldigingen kan worden gedaan.

De beschreven algoritme loopt in $O((\log n)^3)$ rekenkundige stappen. SHAMIR [13] brengt dit met enkele kunstgrepen omlaag tot $O(\log n)$. Dit besluit onze schets van het bewijs van Stelling 6. □

Tellen we bit-operaties, dan is het best bekende resultaat veel poverder:

STELLING 7 (POLLARD [10]). *Voor elke* $\varepsilon > 0$ *bestaat er een algoritme die elk geheel getal* $n > 1$ *volledig factorizeert in* $O(n^{(1/4)+\varepsilon})$ *stappen.*

BEWIJS. Het vereenvoudigde bewijs dat we hier schetsen is afkomstig van V. STRASSEN [14]. Om n te factorizeren is het voldoende zijn factoren $\leq \sqrt{n}$ te kennen, en evenals in het bewijs van Stelling 6 kan men

inzien dat men deze kan bepalen als men weet hoe a! mod n te berekenen, voor $a \leq \sqrt{n}$. De stelling is dus bewezen als we aantonen dat a! mod n in $O(a^{1/2} \cdot n^{\varepsilon})$ stappen bepaald kan worden.

We mogen aannemen dat a een kwadraat is: $a = b^2$. Definieer

$$f(x) = \prod_{i=1}^{b} (x+i),$$

dan geldt

$$a! = \prod_{j=0}^{b-1} f(jb).$$

Uit algemene resultaten over polynoom-arithmetiek volgt dat de coëfficiënten van het polynoom f in $O(b \cdot n^{\varepsilon})$ stappen modulo n berekend kunnen worden, en dat, gegeven f, de b waarden $f(0), f(b), \ldots, f((b-1)b)$ ook in $O(b \cdot n^{\varepsilon})$ stappen modulo n berekend kunnen worden. We kunnen, tenslotte, deze b waarden in $O(b \cdot n^{\varepsilon})$ stappen modulo n vermenigvuldigen, hetgeen a! mod n oplevert. Hiermee is Stelling 7 bewezen. □

Met de Riemann-hypothese gaat het een beetje sneller:

STELLING 8. *Voor elke* $\varepsilon > 0$ *bestaat er een algoritme die elk geheel getal* $n > 1$ *volledig factoriseert en, als de gegeneraliseerde Riemann-hypothese waar is, termineert in* $O(n^{(1/5)+\varepsilon})$ *stappen.*

Men heeft, voor het bewijs van Stelling 8, de keuze uit verschillende algoritmen. Deze maken alle gebruik van de door *Gauss* (1801) ontwikkelde theorie van binaire quadratische vormen $aX^2 + bXY + cY^2$ met gehele coëfficiënten a, b, c, en meer in het bijzonder van het verband dat er bestaat tussen de zogenaamde *ambigue* vormen, waarvoor b deelbaar is door a, en de ontbindingen van de discriminant $\Delta = b^2 - 4ac$ in twee factoren. Hierbij kan men zowel met positieve als met negatieve Δ werken. Voor meer informatie over deze methoden, waarvan de grondgedachte afkomstig is van *Shanks*, zie men de bijdrage van R.J. Schoof in [4].

De Riemann-hypothese die in Stelling 8 bedoeld wordt is algemener dan die uit Stelling 4, en speelt bovendien een andere rol: het is niet de correctheid van het antwoord die erdoor gegarandeerd wordt, maar het feit dat de algoritme binnen de gestelde rekentijd termineert.

Een wezenlijk verschil tussen de problemen (a) (primaliteit) en (b) (factorizatie) bestaat uit het volgende. Bij (a) "weet" men vrij snel het antwoord op de vraag of n priem is, de grote moeilijkheid bestaat eruit

de *juistheid* van dit antwoord te bewijzen. Bij (b) is het echter juist de grote kunst om factoren a,b > 1 van n te *vinden*; als ze eenmaal gevonden zijn is het triviaal om te controleren dat a·b = n. Deze omstandigheid leidt, bij probleem (b), tot een grotere nadruk op algoritmen van probabilistische aard, waarvan het termineren binnen een bepaalde tijd niet gegarandeerd wordt, maar wel op grond van een probabilistische redenering *verwacht* wordt. Verscheidene van de beste in de practijk gebruikte methoden zijn van deze soort; we noemen in het bijzonder de rho-methode van *Pollard* en de "square-form factorization"-algoritme (SQUFOF) van *Shanks*, die beide verwachte rekentijd $O(n^{(1/4)+\varepsilon})$ hebben. Hier willen we vooral aandacht besteden aan methoden waarvan de verwachte rekentijd sneller dan elke macht van n is. Zie SCHNORR [12] voor een uitgebreidere behandeling.

De te bespreken methoden beginnen ermee een reeks paren gehele getallen (c_i,d_i) te construeren waarvoor geldt

$$c_i^2 \equiv d_i \bmod n,$$

alle priemfactoren van $|d_i|$ zijn "klein", zeg $\leq B$.

Iedere d_i ontbindt men in priemfactoren:

$$d_i = (-1)^{n_{i1}} \cdot \prod_{p \text{ priem},\ p\leq B} p^{n_{ip}}.$$

Van de vectoren $(n_{i1},n_{i2},n_{i3},n_{i5},\ldots)$ beschouwt men nu de coördinaten modulo 2. Heeft men genoeg paren (c_i,d_i) dan kan men met behulp van lineaire algebra over $\mathbb{Z}/2\mathbb{Z}$ een relatie tussen deze vectoren vinden:

$$\sum_{i\in I} n_{ip} \equiv 0 \bmod 2 \quad \text{voor } p = 1,2,3,5,\ldots .$$

Dan is $\prod_{i\in I} d_i$ een *kwadraat*, zeg e^2, en met $g \equiv \prod_{i\in I} c_i \bmod n$ geldt dan

$$g^2 \equiv e^2 \bmod n,$$

dus n is een deler van $g^2 - e^2 = (g-e)\cdot(g+e)$. Nu hoopt men dat $g \not\equiv \pm e \bmod n$, dan zijn ggd(n,g-e) en ggd(n,g+e) niet-triviale factoren van n.

De diverse algoritmen die op het bovenstaande principe berusten verschillen voornamelijk in de manier waarop de paren (c_i,d_i) gegenereerd worden.

Het eenvoudigst gaat DIXON [3] te werk. Hij kiest de c_i willekeurig uit $\{1,2,\ldots,n\}$, zet $d_i = (c_i^2 \bmod n)$, en verwerpt de paren (c_i,d_i) waarvoor d_i

een priemfactor > B heeft, met $B = \exp(\sqrt{2 \log n \operatorname{loglog} n})$. Dixon is dan in staat een stelling te formuleren en te bewijzen, die ruwweg zegt dat men mag verwachten dat de algoritme in $O(B^3) = O(\exp(3\sqrt{2 \log n \operatorname{loglog} n}))$ stappen termineert. Uit

$$\exp(\sqrt{\log n \operatorname{loglog} n}) = n^{\sqrt{\operatorname{loglog} n / \log n}} = (\log n)^{\sqrt{\log n / \operatorname{loglog} n}}$$

blijkt dat dit sneller dan elke vaste macht van n is, maar langzamer dan elke macht van log n. Het geheugengebruik van Dixon's methode is $O(B^2)$.

Een tweede, en in de practijk ook werkelijk gebruikte methode om de paren (c_i, d_i) te genereren maakt gebruik van de kettingbreukontwikkeling van $\sqrt{n}$ (of $\sqrt{kn}$, met k klein). De kettingbreukontwikkeling levert rationale benaderingen A_i/B_i van $\sqrt{n}$, en men neemt $c_i = A_i \bmod n$, $d_i \equiv A_i^2 \bmod n$, $|d_i|$ minimaal. Omdat A_i "dichtbij" $B_i\sqrt{n}$ ligt, ligt A_i^2 "dichtbij" het veelvoud $B_i^2 n$ van n, dus men verwacht dat d_i "klein" is, en inderdaad kan men bewijzen

$$|d_i| < 2\sqrt{n}.$$

Dit verkleint de kans dat een paar (c_i, d_i) verworpen moet worden vanwege een te grote priemfactor in d_i. Met een heuristische redenering heeft *Wunderlich* laten zien dat men met $B = \exp(\sqrt{(1/3) \log n \operatorname{loglog} n})$ terminatie in tijd $O(B^3) = \exp(\sqrt{3 \log n \operatorname{loglog} n})$ mag verwachten. Dit is sneller dan *Dixon* maar er is hier geen sprake van een precieze stelling.

De net beschreven methode werd door *Lehmer* en *Powers* voorgesteld in 1931, de tijd dat men nog met de hand rekende. Er bleek een niet te verwaarlozen kans te bestaan dat men, hopende bijna klaar te zijn, ontdekt dat $g \equiv \pm e \bmod n$, zodat men practisch weer helemaal opnieuw moet beginnen. Door dit frustrerende effect werd de methode niet populair. *Brillhart* en *Morrison* realiseerden zich dat een computer niet gevoelig is voor dit nadeel van de methode, en haalden hem in 1970 weer van stal. Voor erg grote getallen (30 à 40 cijfers) is het de beste in de practijk gebruikte algoritme.

Tenslotte is er een algoritme van *Schroeppel*, die dezelfde grondgedachte iets anders uitwerkt. In plaats van met congruenties $c_i^2 \equiv d_i \bmod n$ werkt hij met congruenties

$$\prod_{c \in C} c^{m_{ic}} \equiv d_i \bmod n$$

$$m_{ic} \in \mathbb{Z}_{\geq 0}, \quad \sum_{c \in C} m_{ic} = 2,$$

waarbij C een vaste collectie getallen in de buurt van $\sqrt{n}$ is, zodat d_i weer klein is. Men kiest het aantal elementen van C gelijk aan het aantal priemgetallen $\leq B$. De vectoren $(n_{i1}, n_{i2}, \ldots)$ maakt men 2× zo lang door er de m_{ic} aan toe te voegen. Dit leidt dan weer op dezelfde manier tot een congruentie $g^2 \equiv e^2 \bmod n$.

Schroeppel geeft de verzameling C een speciale structuur, waardoor het tijdrovende factorizeren van $d_i = (\prod c^{m_{ic}}) - n$ kan geschieden door te zeven. Verder beredeneert hij dat $B = \exp(\frac{1}{2}\sqrt{\log n \, \operatorname{loglog} n})$ de optimale keuze is. Dit leidt tot een algoritme met verwachte rekentijd $O(B^3) = \exp(\frac{3}{2}\sqrt{\log n \, \operatorname{loglog} n})$ (de in een ongepubliceerd geschrift van *Boonstra* genoemde grens $O(B^2)$, die Schroeppel claimt, wordt niet door de daar genoemde argumenten ondersteund). Opnieuw is hier geen sprake van een precies geformuleerde en bewezen stelling.

Samenvattend kunnen we concluderen dat probleem (a) "doenlijk" is, vooral als men aan een morele zekerheid over de juistheid van het antwoord voldoende heeft, maar dat probleem (b) met de tegenwoordige kennis "ondoenlijk" is voor algemene n.

2. LITERATUUR

[1] ADLEMAN, L.M., *On distinguishing prime numbers from composite numbers* (abstract), Proc. 21st Annual IEEE Symp. Found. Comp. Sci. (1980), 387-406.

[2] ADLEMAN, L.M., C. POMERANCE & R.S. RUMELY, *On distinguishing prime numbers from composite numbers*, Preprint.

[3] DIXON, J.D., *Asymptotically fast factorization of integers*, Math. Comp. 36 (1981), 255-260.

[4] GETALTHEORIE EN COMPUTERS, Studieweek, Mathematisch Centrum 1980.

[5] GUY, R.K., *How to factor a number*, Proc. Fifth Manitoba Conf. Numer. Math, Utilitas, Winnipeg (1975), 49-89.

[6] KNUTH, D.E., *The art of computer programming, vol. 2*, Seminumerical Algorithms, second edition, Addison-Wesley, Reading 1980.

[7] LENSTRA, H.W., JR., *Primality testing algorithms*, Séminaire Bourbaki 33 (1980/81), exp. 576, Lecture Notes in Mathematics, Springer, to

appear.

[8] MILLER, G.L., *Riemann's hypothesis and tests for primality,* J. Comput. System Sci. 13 (1976), 300-317.

[9] MONIER, L., *Algorithmes de factorization d'entiers,* Thèse, Orsay 1980.

[10] POLLARD, J.M., *Theorems on factorization and primality testing,* Proc. Cambridge Philos. Soc. 76 (1974), 521-528.

[11] PRATT, V.R., *Every prime has a succinct certificate,* SIAM J. Comput. 4 (1975), 214-220.

[12] SCHNORR, C.P., *Refined analysis and improvements on some factoring algorithms,* to appear in: Automata, Languages and Programming, Eighth Colloquium, Haifa 1981, Lecture Notes in Computer Science, Springer.

[13] SHAMIR, A., *Factoring numbers in* O(log n) *arithmetic steps,* Inform. Process. Lett. 8 (1979), 28-31.

[14] STRASSEN, V., *Einige Resultate uber Berechnungskomplexität,* Jber. Deutsche Math. Verein. 78 (1976), 1-8.

[15] WILLIAMS, M.C., *Primality testing on a computer,* Ars Combin. 5 (1978), 127-185.

EFFICIËNTIE VERSUS NAUWKEURIGHEID

C.G. VAN DER LAAN

0. INLEIDING

De laatste jaren is er heel wat verschenen op het gebied van de complexiteitstheorie. Overzichtswerken zijn HARTMANIS & HOPCROFT [21], MILLER & THATCHER [34], HOPCROFT [24], AHO, HOPCROFT & ULLMAN [1], TRAUB [52,53], BORODIN & MUNRO [5], WINOGRAD [59], en de Turing award lecture: RABIN [38]. Onderscheid wordt gemaakt in analytische complexiteit - de complexiteit om de analytische operator te benaderen door een algebraïsche (RICE in TRAUB [52]), en algebraische of arithmetische complexiteit - de studie van efficiënte of optimale algoritmen voor de evaluatie van algebraïsche functies op (geïdealiseerde) computers (BORODIN & MUNRO [5]). Bij het doorbladeren van bovengenoemde literatuur kan men constateren dat veel aandacht m.b.t. numerieke zaken, is geschonken aan

. operaties op polynomen (evaluatie, vermenigvuldiging, deling);
. matrix-vermenigvuldiging;
. de Discrete Fourier Transformatie.

De aandacht concentreerde zich op het minimale aantal arithmetische operaties, afgezien van geheugengebruik, begrijpbaarheid, parallelle verwerking en ... nauwkeurigheid; dit alles voor *algemene* problemen.
Al deze aspecten kunnen tot gevolg hebben dat niet de algoritme met de minimale complexiteit gebruikt zal worden, omdat bijvoorbeeld het geheugenaccess te veel tijd vergt, het programma niet meer begrijpbaar is, de sequentiële verwerking in vergelijking met parallelle verwerking te lang duurt, of helaas de algoritme *ontoelaatbaar* onnauwkeurige resultaten levert. Verder is er in een concrete situatie veelal extra probleeminformatie voor handen, waardoor de complexiteit van het specifieke probleem een orde kleiner is dan de complexiteit van het algemene probleem. Met het opkomen van de vector-computers, array-processors e.d. is het ontwerpen van parallelle

algoritmen van practisch belang. Algoritmen gebaseerd op de *divide-and-conquer* techniek lenen zich van nature voor parallelle verwerking. Een bibliografie t.a.v. parallelle algoritmen is gegeven door POOLE & VOIGT [36].

OPMERKING. Parallelle verwerking betekent niet dat de complexiteit kleiner is, slechts de *tijdsduur* wordt beïnvloed omdat een aantal bewerkingen tegelijkertijd gebeuren. Hierdoor kan men algoritmen beschouwen die een grotere complexiteit hebben, maar waarbij de parallelle verwerking sneller is dan de sequentiële verwerking.

Bovendien moet men in ogenschouw nemen dat men analoog aan het *vooraf* verrichten van wiskundige manipulaties ter 'vereenvoudiging', men ook een precomputing fase kan onderscheiden. Dit is met name van belang bij uitspraken over de optimaliteit van het Hornerschema, zoals gedaan door Vitanyi (deze syllabus), waarbij wij na *precomputing* tot efficiëntere representaties kunnen komen (zie 2.1). Dit is van belang bij evaluatie van een polynoom voor meerdere waarden van het argument.

De numerieke wiskunde houdt zich bezig met onderzoek naar efficiënte (zowel ten aanzien van runtime als geheugenbeslag) én stabiele algoritmen, voor problemen die voortkomen uit: ... analysis of mathematical models of the physical world and the optimization of models of the organizational world. (RICE c.s. [42]). Het vakgebied is nogal uitgebreid; naast de bibliografie van GINSBURG [19], die verwijzingen bevat naar vele introducties in de numerieke wiskunde, en boeken die gewijd zijn aan één probleemgebied of zelfs aan één probleem is er verschenen: het COSER's rapport (RICE c.s. [42]) en het State-of-the-Art boek JACOBS [27]. Ten aanzien van de foutenanalyse en de stabiliteit is er WILKINSON [57,58], HOUSEHOLDER [25], STERBENZ [48], RUTISHAUSER [45], KAHAN [28], BAUER [4] en BABUSKA [2]; voor foutenanalyse m.b.v. een computer is er MILLER [32,33]. Verder is er een discussie gaande over de standaardisatie van computer arithmetiek.

Zowel vanuit de complexiteitstheorie als vanuit de numerieke wiskunde wordt er gekwantificeerd. De efficiëntie en nauwkeurigheid moeten meetbaar zijn om tegen elkaar afgewogen te kunnen worden. In beide groeperingen echter zijn er relativerende geluiden te horen; het openstaan voor de nevenaspecten tijdens het zoeken naar optimale grenzen wordt belangrijk geacht. ... There is still a tendency to attach too much importance to the precise

error bounds obtained by an apriori error analysis. In my opinion the bound itself is usually the least important part of it. The main object of such an analysis is to expose the potential instabilities, if any, of an algorithm so that hopefully from the insight thus obtained one may be led to improved algorithms. Usually the bound itself is weaker than it might have been because of the necessity of restricting the mass of detail to a reasonable level and because of the limitations imposed by expressing the errors in terms of matrix norms. A priori bounds are not, in general, quantities that should be used in practice. Practical error bounds should usually be determined by some form of a posteriori error analysis, since this takes full advantage of the statistical distribution of rounding errors and of any special features, such as sparseness in the matrix ... (WILKINSON [57]). ... The view we take here is an attempt to understand what makes functions hard to compute, rather than to provide the best practical algorithm. This view leads to a concern with asymptotic behavior and the presentation of some algorithms simply as 'proofs of upper bounds', not as guidelines for practical computation. ... (BORODIN & MUNRO [5]).

1. NAUWKEURIGHEID

De nauwkeurigheid van het resultaat van een algebraïsche operator als benadering van een analytische operator wordt bepaald door:

. het effect van verstoringen van de operand;
. de approximatie (residue of truncatie) fout;
. het effect van de (eindige precisie) arithmetiek.

Zij

$f(x)$: de waarde van de analytische operator met argument x,
$\tilde{f}(x;a)$: de waarde van de benaderende algebraïsche operator met parameters a,
$\tilde{f}_p(\tilde{x};a)$: het resultaat van $\tilde{f}$ via algoritme p in eindige precisie met verstoorde operand $\tilde{x}$,

dan geldt voor de nauwkeurigheid

$$|f(x) - \tilde{f}_p(\tilde{x};a)| \leq |f(x) - f(\tilde{x})| + |f(\tilde{x}) - \tilde{f}(\tilde{x};a)| + |\tilde{f}(\tilde{x};a) - \tilde{f}_p(\tilde{x};a)|.$$

Bij een a priori analyse kunnen wij ernaar streven die benadering $\tilde{f}$ en algoritme p te kiezen zodat de tweede én derde term klein genoeg zijn. Bij het zoeken naar de algebraïsche benadering hebben wij vrijheid van representatie d.w.z. wij kunnen zowel $\tilde{f}(\tilde{x};a)$ als $\tilde{g}(\tilde{x};b)$ kiezen als voor beiden

geldt

$$|f(\tilde{x}) - \tilde{f}(\tilde{x};a)| < \text{truncatiefout}$$
$$|f(\tilde{x}) - \tilde{g}(x;b)| < \text{truncatiefout}.$$

Een bijzonder geval is de verschillende representatie: $\tilde{f}(\tilde{x};a) = \tilde{g}(\tilde{x};b)$.
De binnen de truncatiefout geboden vrijheid kunnen wij benutten door een benadering te kiezen met zowel een zo goed mogelijke *conditie* als een zo goed mogelijke analytische complexiteit.
Bij het kiezen van de algoritme p kunnen wij onze keuze laten bepalen door een zo klein mogelijke *groeifactor* en een zo klein mogelijke algebraïsche complexiteit.
Wij hebben dus te maken met de begrippen: analytische en algebraïsche complexiteit versus conditie en groeifactor. Wij kunnen ons nu verschillende werkwijzen t.a.v. het construeren van algoritmen voorstellen:

. De *robuuste numericus* die eenmalig een probleem moet oplossen.
  Deze zou kunnen afzien van algoritmen met optimale complexiteit, in zowel algebraïsche als analytische zin, en zich kunnen richten op goed gestelde problemen - goed geconditioneerd t.a.v. verstoringen in de parameters - als benadering, en op algoritmen met geen of kleine groei.
  Bijvoorbeeld: het hanteren van orthogonale transformaties, approximeren met Chebyshevreeksen.
. De *complexicus*.
  Deze zou de gevolgen t.a.v. de eindige precisie kunnen verwaarlozen en zou zich volledig kunnen laten leiden door optimale complexiteit.
  Bijvoorbeeld: operaties op polynomen in machtssomrepresentatie van hoge graad.
. De *programmatuurbouwkundige*.
  Deze moet de nauwkeurigheid én de complexiteit beschouwen naast de aspecten die samenhangen met het ontwikkelen van programmatheken. Dit betekent dat hij moet aangeven: hoe de inherente fout

  $$|f(x) - f(\tilde{x})|$$

  geschat kan worden in termen van $|x - \tilde{x}|$, hoe groot de truncatiefout is (meestal blijft dit achter de schermen) en hoe de gegenereerde fout

  $$|\tilde{f}(\tilde{x};a) - \tilde{f}_p(\tilde{x};a)|$$

geschat kan worden door

conditie * groeifactor * machine precisie

waarbij de groeifactor, hetzij a priori hetzij a posteriori bepaald, als maat voor de verstoringen optreedt.
Hij zou er dus voor moeten zorgen dat het product: conditie * groeifactor, zo klein mogelijk blijft bij het zoeken naar optimale complexiteit. In zogenaamde kritieke stukjes rekenwerk zou hij in multi-lengte kunnen werken als de groeifactor te hoog wordt. Een werkwijze zou kunnen zijn: "houd analytische complexiteit én conditie klein" en vervolgens: "houd algebraïsche complexiteit én groeifactor klein". Als men de mogelijkheid heeft om het onderste uit de kan te halen, dan kan men zich de experimenteer-attitude aanmeten door van diverse implementaties zogenaamde *performance profielen* op te stellen, gebaseerd op een representatieve collectie testproblemen. Dit laatste is doenlijk voor een gebruiker met een concrete klasse van problemen in een bepaalde situatie waarbij hij een keus moet maken uit verschillende voorhanden zijnde programmatuur; er bestaat geen *universeel beste*!

1.1. Conditie

De conditie is een maat voor de gevoeligheid van een operator voor de verstoringen van een operand.
Zij f een operator met operand a dan kunnen wij de conditie C definiëren door

$$C(a) = \lim_{\Delta a \to 0} \frac{\|f(a+\Delta a)-f(a)\|}{\|f(a)\|} \Big/ \frac{\|\Delta a\|}{\|a\|} .$$

Afhankelijk van f werkt men ook wel met eerste orde benaderingen.

OPMERKING. Om "zwakke plekken" in een algoritme op te sporen hanteert men ook wel relatieve afgeleiden van de operand naar het resultaat

$$\frac{a_i}{f} \delta_{a_i} f$$

zodat men weet wat het (eerste orde) effect van een enkele verstoring is, onder de aanname dat er geen overige (eventueel compenserende) verstoringen zijn.

1.2. Groeifactor

Binnen de terugwaartse foutenanalyse van algebraïsche processen heeft de groeifactor - als macroscopische grootheid - een duidelijke plaats gekregen. Het begrip groeifactor heeft te maken met de groei van de tussenresultaten van een algoritme. De tussenresultaten bepalen de grootte van de afrondfouten en derhalve is de groeifactor ook bepalend voor de grootte van de gemaakte afrondfouten. Vroeger werden de tussenresultaten klein gehouden door schaling, vooral noodzakelijk door de vaste-komma-arithmetiek. Toen men overging op glijdende-komma-arithmetiek - met zo goed als constante relatieve fout - had men behoefte aan algoritmen met een "kleine groei". Bij een operator f met operand a uitgevoerd via algoritme p in eindige precisie, gaat de terugwaartse foutenanalyse ervan uit dat er een Δa bestaat zodanig dat geldt

$$f_p(a) \equiv f(a+\Delta a);$$

in woorden: de evaluatie van f via p in eindige precisie is gelijk aan de exacte waarde van f voor een verstoorde operand.
De groeifactor g is dan gedefinieerd door

$$g = \frac{\|\Delta a\|}{\|a\|} / \varepsilon,$$

met ε de (eindige) precisie van de arithmetiek. Helaas is het niet altijd mogelijk om *a priori* een realistische groeifactor te vinden, omdat, naast wat WILKINSON (zie inleiding) stelde, de tussentijdse afrondfouten niet a priori geïnterpreteerd kunnen worden als verstoringen van de begingegevens. Dit komt m.n. voor als dezelfde grootheid op verschillende plaatsen in de algoritme wordt gebruikt. Dan kan men óf dezelfde grootheid op verschillende plaatsen als verschillend beschouwen óf men kan het resultaat als functie van de gemaakte fouten beschouwen en dan ontwikkelen vanuit het exacte resultaat naar de fouten. Ter illustratie beschouwen wij het vinden van de wortels $x_{1,2}$ van de vergelijking

$$x^2 + 2bx + c = 0,$$

via de algoritme

$$\tilde{x}_1 = fl(-(b+sgn(b)\sqrt{b^2-c})), \qquad \tilde{x}_2 = fl(c/\tilde{x}_1)$$

d.w.z. $\exists\delta_i$, zodat geldt

$$\tilde{x}_1 = -(b+sgn(b)\sqrt{(b^2(1+\delta_1)-c)(1+\delta_2)}(1+\delta_3))(1+\delta_4)$$

$$\tilde{x}_2 = c/\tilde{x}_1(1+\delta_5).$$

Interpretatie van $\{\delta_i\}$ als verstoringen van b en c geeft moeilijkheden. Men zou dus in eerste instantie over verschillende b's en c's kunnen praten en later deze kunnen conflueren, of men zou $\tilde{x}_{1,2}$ als functie van $\{\delta_i\}$ kunnen beschouwen en deze dan kunnen ontwikkelen vanuit $\{\delta_i\} = \{0\}$. De laatste aanpak is gepubliceerd door MILLER [33,34] in zijn "software for round-off analysis".
Een andere formulering van het laatste idee is gedaan door BAUER [4] en LARSON [26] in een zgn. computational graph, waarbij het effect van de tussenresultaten wordt beschouwd. KAHAN [28] mengt voorwaartse en terugwaartse foutenanalyse door tussentijdse afrondfouten te interpreteren als verstoringen van de begingegevens én de resultaten. De deskundigen zijn het kennelijk niet eens over de te gebruiken methode; nu eens is de ene methode wat handiger, dan weer 'n andere.

Bij de *a posteriori* (terugwaartse) foutenanalyse kan men uitgaande van de verkregen oplossingen $\tilde{x}_{1,2}$ de vraag stellen voor welke verstoorde vergelijking deze waarden exacte wortels zijn, m.a.w. voor welke δ_b en δ_c geldt:

$$ax^2 + b(1+\delta_b)x + c(1+\delta_c) = 0.$$

Hieruit volgt als oplossing

$$\delta_b = -\frac{a}{b}(\tilde{x}_1+\tilde{x}_2) - 1,$$

$$\delta_c = \frac{a}{c}\tilde{x}_1\tilde{x}_2 - 1.$$

De relatieve fout in de wortels kan benaderd worden - onafhankelijk of δ_b, δ_c via a priori of a posteriori analyse verkregen is - met

$$\begin{pmatrix}\delta x_1\\ \delta x_2\end{pmatrix} = \frac{1}{D}\begin{pmatrix} b & c/x_1\\ -b & -c/x_2\end{pmatrix}\begin{pmatrix}\delta b\\ \delta c\end{pmatrix} + \text{h.o.t.}$$

met D de discriminant. (Als er ook nog meetfouten of conversiefouten in a zijn dan moet men bovenstaande formule uitbreiden.).
Het voorbeeld in KAHAN [28] met

$$a = 47.51, \quad b = 47.45, \quad c = 47.39$$

levert dan voor de algoritmen

1: $x_{1,2} = (-b \pm \sqrt{b^2-4ac})/(2a)$

2: $x_1 = -(b+\text{sgn}(b)\sqrt{b^2-4ac})/(2a)$

$x_2 = c/(ax_1)$

de volgende waarden via de a posteriori verkregen formules voor δb en δc

	δb	δc
algoritme 1	- 1974	99.30
algoritme 2	-2	$2_{10}{}^{-4}$

Algoritme 2 heeft kleinere verstoorde operanden dan algoritme 1!
Invulling van deze resultaten in de bovenstaande formule geeft een schatting - a posteriori - van de onnauwkeurigheid van $x_{1,2}$.

Op grond van het bovenstaande kunnen wij verwachten dat algoritme 1 minder nauwkeurige resultaten geeft dan algoritme 2. Bovendien kunnen wij verschillende vuistregels hieruit destilleren:
. de berekende wortels moeten voldoen aan

$$\tilde{x}_1 + \tilde{x}_2 \approx - b/a$$

$$\tilde{x}_1\tilde{x}_2 \approx c/a;$$

zo niet, dan hebben wij een instabiel algoritme;
. bijna samenvallende wortels geeft aanleiding tot een slecht geconditioneerd

probleem; D is klein (onafhankelijk van de norm van de amplificatie-matrix).

Voor andere analyses van dit "eenvoudige" probleem zie KAHAN [28] of BAUER [4]; LARSON & SAMEH [26] hebben Bauer's idee nader uitgewerkt.

OPMERKING. Het aantal benodigde bewerkingen voor de beide algoritmen verschilt nauwelijks.

2. APPROXIMATIE VAN FUNCTIES

Reeds in de beginperiode van de automatische rekenmachine was het efficiënt representeren van standaardfuncties een belangrijke activiteit. Het opslaan van tabellen was duur ten aanzien van de benodigde geheugenruimte en duur t.a.v. de tijd nodig voor het opzoeken van de tabelwaarden en voor het berekenen van de benodigde interpolatie. Al vroeg besloot men daarom de functie te benaderen door eenvoudiger functies, en uiteindelijk door algebraïsche functies. In verband met de verlangde efficiënte en nauwkeurigheid wordt het interval verdeeld. Vervolgens kan men op ieder deelinterval een zo efficiënt mogelijke benadering construeren, of uitgaande van de benadering op een deelinterval via bijvoorbeeld recursie de functie benaderen op het andere interval. Als wij ons nu concentreren op het deelprobleem van het benaderen van de functie f op een geschikt gekozen deelinterval, dan hebben wij te maken met

$$\tilde{f}(x;a) \quad \text{en} \quad \tilde{f}_p(x;a),$$

waarbij wij ook de transformatie van de afhankelijke en onafhankelijke variabele nog kunnen beschouwen.

Als voorbeelden van $\tilde{f}$ zou men kunnen denken aan
. een polynoom of rationale functie als minimax benadering;
. een reeks (Chebyshev, Taylor, asymptotisch);
. een kettingbreuk,
al dan niet gecombineerd met recursie.

De analytische complexiteit, bijvoorbeeld het aantal benodigde termen in een reeks, is afhankelijk van de gewenste nauwkeurigheid en de grootte van het interval. (Het is dan ook gemakkelijk te begrijpen dat bij te dure benaderingen men simpelweg het benaderend interval in stukken verdeelt). Bij

het selecteren van de benadering is het raadzaam m.b.t. het vergelijken van de conditie de absolute som van de relatieve afgeleiden naar de parameters te vergelijken

$$\sum_k \left| \frac{a_k}{\tilde{f}} \frac{\partial \tilde{f}(x;a)}{\partial a_k} \right| .$$

VOORBEELDEN (conditie van de approximerende vorm)

1. Zij

$$e^{-x} = \sum_{k=0}^{\infty} (-1)^k x^k/k! = 1/ \sum_{k=0}^{\infty} x^k/k!, \qquad x \in [0,x_0]$$

dan geldt voor de bovenstaande absolute som voor beide reeksen

$$e^{2x_0} \text{ en } 1,$$

op grond waarvan wij de eerste representatie als numeriek niet goed genoeg geconditioneerd kunnen verwerpen. (WILKINSON [58, p.36,37] moet meer werk doen om dit aan te tonen).

2. Zij

$$p_5(x) = \sum_{k=0}^{5} a_k x^k = 1 - 13.7x + 67.5x^2 - 153x^2 + 162x^4 - 64.8x^5$$

$$= \sum_{k=0}^{2} b_{2k+1} T^*_{2k+1}(x) = -(.522T^*_1(x) + .352T^*_3(x) + .126T^*_5(x))$$

met $T^*_k(x)$ het verschoven Chebyshevpolynoom, dan hebben wij voor de bovenstaande absolute sommen voor $x = 1$

$$462 \text{ respectievelijk } 1,$$

op grond waarvan wij de eerste representatie als numeriek niet goed genoeg geconditioneerd kunnen verwerpen (RUTISHAUSER [44]).

3. Ook bij een kettingbreuk hebben wij een diversiteit aan equivalente representaties met de even en oneven contractie als bekende voorbeelden.

OPMERKINGEN. Vervolgens moeten wij een algoritme kiezen die zowel een zo klein mogelijke algebraïsche complexiteit als een zo klein mogelijke groei heeft (zie 2.1 t.a.v. polynomen).

. Het argument x is niet beschouwd in de relatieve afgeleiden omdat deze te maken heeft met de conditie van de functie en niet zo zeer met de conditie van de benaderingsfunctie. Soms kan het echter nuttig zijn een transformatie van de afhankelijke en/of onafhankelijke variabele toe te passen om daarmee een efficiëntere benadering dus een kleinere analytische complexiteit - van f te verkrijgen:

$$f(x) = h(g(x)) \qquad \text{(transformatie afhankelijke variabele)}$$

$$= f(x(t)) \qquad \text{(transformatie onafhankelijke variabele)}$$

$$= h(v(t)) \qquad \text{(transformatie afhankelijke en onafhankelijke variabele).}$$

VOORBEELDEN. (TEMME en VAN DER LAAN [51])

1. In de NAG-programmatheek gebruikt SCHONFELDER [46]

$$E_1(x) = e^{-x}/x\, e_1(t), \qquad x \in [4,xhi]$$

met

$$t = (11.25-x)/(3.25+x)$$

waarbij de Chebyshevreeks van $e_1(t)$ sneller convergeert dan de Chebyshevreeks van $xe^{x}E_1(x)$; de analytische complexiteit is verkleind. Bij een relatieve fout van $\sim 10^{-15}$ zijn voor de intervallen (0,4) en (4,∞), 16 respectievelijk 15 termen van de Chebyshevreeks nodig. De benadering is om efficiëntie redenen a priori getransformeerd naar een polynoom in de machtssomrepresentatie terwijl de stabiliteit behouden bleef; de algebraïsche complexiteit is verkleind, onder behoud van de nauwkeurigheid (experimenteel).

2. De exponentiële integraal $E_1(x)$ is te representeren door de reeks

$$E_1(x) = -\ln(x) - \gamma - \sum_{m=1}^{\infty} \frac{(-x)^n}{nn!}, \qquad x \in \mathbb{R}^+$$

en door de Legendre kettingbreuk

$$E_1(x) = e^{-x}\left(\frac{1}{x+}\,\frac{n}{1+}\,\frac{n+1}{1+}\,\frac{2}{x+}\ldots\right), \qquad x \in \mathbb{R}^+.$$

Uitgaande van deze in principe goede benaderingen hebben CODY & THACHER

[7] efficiënte, a posteriori stabiele, rationale minimax benaderingen geconstrueerd van de "wiskundig rustiger" functies

$$E_1(x) + \ln(x) \quad \text{op} \quad (0,1]$$
$$e^x E_1(x) \quad \text{op} \quad [1,4]$$
$$x^2 e^x E_1(x) - x \quad \text{op} \quad [4,\infty).$$

Als tussenfase hebben zij de reeks in een kettingbreuk getransformeerd via de QD-algoritme; deze kettingbreuk werd op (0,4] als moederfunctie gebruikt. Uitgaande van de wens dat de benaderende rationale functies van dezelfde complexiteit op de verschillende intervallen moesten zijn, bleek empirisch de opsplitsing in (0,1], [1,4], [4,∞) te voldoen. Bovendien bleek de efficiëntere J-fractie voor bepaalde gebieden de meest stabiele. Bij een relatieve fout van $\sim 10^{-15}$ is de graad van de teller en noemer van de rationale functie ± 6 afhankelijk van het interval. Op [1,4] werd de teller en noemer a priori gedeeld door de hoogste macht van x! Bij hun analoge werk voor Ei(x) verkregen zij rationale functies die zij empirisch niet nauwkeurig genoeg vonden; de representatie van de teller en noemer in verschoven Chebyshevpolynomen gaf voldoende nauwkeurige resultaten.

OPMERKING. Zowel Schonfelder als Cody & Thacher hebben kennelijk voor de gebruikte polynomen de *streamlined* of *adapted* representaties niet beschouwd.

3. De sinus- en cosinus-integralen

$$\mathrm{Si}(x) = \int_0^x \frac{\sin t}{t}\,dt$$
$$\mathrm{Ci}(x) = \gamma + \ln x + \int_0^x \frac{\cos t - 1}{t}\,dt$$

kunnen gerepresenteerd worden via

$$\begin{pmatrix} \mathrm{Si}(x) \\ \mathrm{Ci}(x) \end{pmatrix} = \begin{pmatrix} \pi/2 \\ 0 \end{pmatrix} - \begin{pmatrix} \cos x & \sin x \\ -\sin x & \cos x \end{pmatrix} \begin{pmatrix} f(x) \\ g(x) \end{pmatrix}, \qquad x \neq 0$$

met

$$f(x) = \int_0^\infty \frac{\sin t}{t+x}\,dt$$

$$g(x) = \int_0^\infty \frac{\cos t}{t+x}\,dt.$$

Door het slingerend gedrag af te splitsen hoeven nu slechts de "wiskundig rustiger" functies f en g benaderd te worden: de analytische complexiteit is verkleind, voor grote waarden van x.
SCHONFELDER [46], geïnspireerd door BULIRSCH, verkleind de analytische complexiteit verder door

$$xf(x) \quad \text{en} \quad x^2 g(x), \qquad 16 \le x < xhi$$

te ontwikkelen in Chebyshev reeksen

$$\sum a_k T_k(t), \qquad t = 2(16/x)^2 - 1,$$

(voor $0 < x < 16$ worden

$$Si(x)/x \quad \text{en} \quad Ci(x) - \ln(x)$$

ontwikkeld in Chebyshev reeksen

$$\sum a_k T_k(t), \qquad t = 2(x/16)^2 - 1).$$

2.1. Evaluatie van een polynoom

STELLING (KNUTH [30,p.435])
Iedere n-de *graads polynoom,* u(x), $n \ge 3$, *met reële coëfficiënten, en leidende coëfficient* u_n, *kan geevalueerd worden via het schema*

$$u(x) = (\ldots((\beta_0'(w-\alpha_1)+\beta_1)(w-\alpha_2)+\beta_2)\ldots)(w-\alpha_m)+\beta_m$$

met

$$m = \lceil n/2 \rceil - 1,$$

$$\beta_0' = \begin{cases} (u_n y+\alpha_0)y + \beta_0, & \text{als n is even} \\ u_n y + \beta_0, & \text{als n is oneven} \end{cases}$$

en

$$y = x + c, \qquad w = y^2,$$

$$\alpha_k, \beta_k, c \in \mathbb{R}.$$

Bovendien kunnen de parameters zo bepaald worden dat $\beta_m = 0$.

Uit bovenstaande representatie volgt dat de evaluatie n additities en $\lfloor n/2 \rfloor + 2$ vermenigvuldigingen vraagt. Het aantal wiskundig equivalente representaties is minstens $2 \times (m-1)!$; onduidelijk is welke numeriek te verkiezen is. (Voor n = 4,5,6 heeft KNUTH [30,p.432,433] efficiëntere representaties gegeven, die terugvoeren op PAN en MOTZKIN & BELAGA).

Het merkwaardige is dat nergens de absolute som van de relatieve afgeleiden naar de parameters is vergeleken. RICE [43] heeft experimenteel de nauwkeurigheid van een aantal van deze "streamlined" vormen vergeleken met het Hornerschema; ruwweg de helft van de "streamlined" (RICE) of "adapted" (KNUTH) representaties waren minder nauwkeurig. CODY [9] memoreert het geheugenacces als een extra factor. Deze resultaten suggereren dat men voor elke concrete situatie voor de verschillende vormen een nauwkeurigheidsprofiel moet opstellen op grond waarvan een keuze gemaakt kan worden.

Foutenanalyse Hornerschema

Het polynoom

$$P_n(\{a_k\};x) = \sum_{k=0}^{n} a_k x^k$$

uitgerekend in eindige precisie geeft

$$fl\{P_n(\{a_k\};x)\} \equiv P_n(\{a_k(1+\delta_k)\};x)$$

met voor $k < n$

$$(1-\varepsilon_0)\prod_{j=1}^{k}((1-\varepsilon_j)(1-\varepsilon_j')) \le 1 + \delta_k \le (1+\varepsilon_0)\prod_{j=1}^{k}((1+\varepsilon_j)(1+\varepsilon_j')).$$

Hieruit volgt in eerste orde

$$|\delta_k| < (2k+1)\varepsilon, \ \varepsilon = \max_j(\varepsilon_j, \varepsilon_j'), \qquad |\delta_n| < 2n\varepsilon.$$

Als schatting van de fout kunnen wij hanteren

$$|P_n(\{a_k\delta_k\};x)| \leq \varepsilon\{P_n(\{|a_k|\};|x|) + 2|x|P_n'(\{|a_k|\};|x|)\}$$

$$\leq 2n\varepsilon P_n(\{|a_k|\},|x|),$$

waarbij $|P_n(\{|a_k|\},|x|)/P_n(\{a_k\};x)|$ als conditie opgevat kan worden en 2n als "overschatting" van de groei.

Foutenanalyse "adapted" vorm

Het polynoom $P_n(\{a_k\};x)$ in de representatie

$$Q_m(\{\beta_k\};\{w-\alpha_k\}) = \sum_{k=0}^{m} \beta_k \prod_{j=k+1}^{m} (w-\alpha_j)$$

uitgerekend in eindige precisie geeft

$$fl\{Q_m(\{\beta_k\};\{w-\alpha_k\})\} \equiv Q_m(\{\beta_k(1+\delta_k)\};\{(w-\alpha_k)(1+\eta_k)\})$$

met

$$1 - \varepsilon_k \leq 1 + \delta_k \leq 1 + \varepsilon_k, \qquad k \geq 0$$

$$(1-\varepsilon_k)(1-\varepsilon_k') \leq 1 + \eta_k \leq (1+\varepsilon_k)(1+\varepsilon_k'), \qquad k > 0$$

waarbij wij de verstoringen in u_n (zie p.13) verwaarlozen.
Als schatting van de fout kunnen wij hanteren

$$|Q_m(\{\beta_k\xi_k\};\{w-\alpha_k\})|,$$

met -

$$\xi_k = \delta_k + \left(\prod_{j=k+1}^{m} (1+\eta_j)-1\right) \text{ en } |\xi_k| \lesssim (2(m-k)+1)\varepsilon .$$

Een bovengrens voor de fout is

$$2m\varepsilon Q_m(\{|\beta_k|\};\{|w-\alpha_k|\})$$

waarbij

$$Q_m(\{|\beta_k|\};\ \{|w-\alpha_k|\ )/|P_n(\{a_k\};\ x)|$$

als conditie (t.a.v. de nieuwe parameters) en 2m als overschatting van de groei opgevat kunnen worden.

Op grond van bovenstaande analyses zouden wij de conditiefuncties van de verschillende representaties kunnen gaan vergelijken voor een aantal waarden van het argument; op grond daarvan zouden wij die representatie kunnen kiezen met de kleinste *bovengrens* van de fout, hetgeen niet behoeft te betekenen dat de feitelijke fout het kleinst is.

VOORBEELDEN (Horner versus "adapted" vorm)

1. Voor de evaluatie van het 5e graads polynoom uit KNUTH [30,Ch.4.6.4,exc. 19] zijn er de representaties

   Horner: $u_5(x) = (...(x+5)x-10)x-50)x+13)x + 60$

   en

   "adapted" vorm: $u_5(x) = ((x^2-10)x^2+13)(x+5) - 5.$

   Een indruk van de relatieve conditie kunnen wij verkrijgen door de bovengenoemde conditiefuncties te vergelijken; voor x = 1 verkregen wij

   Horner : $139/|p_5(1)|$

   "adapted": $137/|p_5(1)|$

   op grond waarvan de "adapted" vorm te verkiezen is, omdat deze efficiënter is en geen grotere grens voor de fout heeft. (De bovenstaande "adapted" vorm is een van de drie mogelijke vormen).

2. Voor de evaluatie van het 6e graads polynoom uit KNUTH [30,Ch.4.6.4,p.433] gebruiken wij

   Horner: $u_6(x) = (...(x+13)x+49)x+33)x-61)x-37)x + 3$

   en

"adapted" vorm: $z = (x+3)x - 7$

$w = (x+3)z + 16$

$u_6(x) = (w+z+6)w - 27.$

De absolute som der relatieve afgeleiden naar de parameters, i.v.m. de relatieve conditie, gaf voor x = 1 voor Horner: $197/|u_6(1)|$ en voor de "adapted" vorm $576/|u_6(1)|$. De absolute som der relatieve afgeleiden naar de tussenresultaten gaf voor x = 1 voor Horner: $355/|u_6(1)|$ en voor de "adapted" vorm $468/|u_6(1)|$.

2.2. Evaluatie van een rationale functie

KNUTH [30,p.439] vermeldt dat het analogon van Horner's regel voor rationale functies - de zogenaamde J-fractie, RICE [43] - de optimale vorm is m.b.t. het aantal evaluaties, als vermenigvuldiging en deling vergelijkbaar zijn. (Een ALGOL 60 programma voor conversie naar een J-fractie, en omgekeerd, is gepubliceerd in HART [20].)

De J-fractie behorend bij een rationale functie, met in de teller en de noemer polynomen van de graad n, is gedefinieerd door

$$J(x;\alpha,\beta) = \alpha_0 + \frac{\alpha_1 \mid}{\beta_1+x+} \frac{\alpha_2 \mid}{\beta_2+x+} \cdots \frac{\alpha_n \mid}{\beta_n+x} = \alpha_0 \overset{n}{+_{k=1}} \frac{\alpha_k}{\beta_k+x} \quad .$$

De evaluatie vraagt n delingen en 2n optellingen. Als wij nu definiëren

$$r_j = \frac{\alpha_j}{\beta_j+x+r_{j+1}}, \qquad j = 1,2,\ldots,n,$$

$$r_{n+1} = 0$$

dan geldt

$$J(x;\alpha,\beta) = \alpha_0 + r_1$$

waarbij de relatieve afgeleiden kunnen worden gerepresenteerd door

$$\frac{\alpha_0}{J}\frac{\partial J}{\partial \alpha_0} = \frac{\alpha_0}{J}$$

en voor k = 1,2,...,n

$$\frac{\alpha_k}{J}\frac{\partial J}{\partial \alpha_k} = \frac{r_k}{J}\prod_{j=1}^{k-1}\left(-\frac{r_j^2}{\alpha_j}\right)$$

$$\frac{\beta_k}{J}\frac{\partial J}{\partial \beta_k} = \frac{\beta_k}{J}\prod_{j=1}^{k}\left(-\frac{r_j^2}{\alpha_j}\right) .$$

De bovenstaande recursie in eindige precisie kan geïnterpreteerd worden als

$$r_j = fl\left(\frac{\alpha_j}{\beta_j+x+r_{j+1}}\right)$$

$$= \frac{\alpha_j(1+\delta_{j3})}{((\beta_j+x)(1+\delta_{j1})+r_{j+1})(1+\delta_{j2})}$$

dus

$$\tilde{\alpha}_j = \alpha_j(1+\delta_{j3})/(1+\delta_{j2})$$

$$\widetilde{(\beta_j+x)} = (\beta_j+x)(1+\delta_{j1});$$

de algoritme heeft dus geen groei!!

VOORBEELD (KNUTH [30,p.443])

$$R(x) = \frac{x^2+10x+29}{x^2+8x+19} = 1 + \frac{2}{x+3-}\,\Big|\,\frac{6}{x+5} .$$

De absolute som van de relatieve afgeleiden van de rationale functie naar de parameters van de polynomen is gelijk aan de absolute som van de relatieve afgeleiden van ieder van de polynomen naar zijn parameters.
Wij krijgen dan voor x = 1 voor Horner: 68/R(1), en voor de J-fractie (2/3)/R(1).

VOORBEELD. CODY & THACHER [8] geven voor de functie $xe^{-x}E_i(x)$ op $6 \le x \le 12$ o.a. de J-fractie benadering

$$J(x) = 9.79202_{10}-1 + \frac{1.24646}{x-1.85524} .$$

De bijbehorende rationale functie luidt

$$\frac{.979202\,x - 1.81665}{x-1.85524} .$$

De absolute som van de relatieve afgeleiden naar de parameters voor x = 6 is voor de J-fractie 1.6/|J(6)| en voor de rationale functie 15.5/|J(6)|, op grond waarvan de J-fractie te prefereren is.

OPMERKINGEN.

. HOLLENBERG [23] beschouwt de relatieve afgeleiden voor de kettingbreuk-representatie

$$\Phi_k \frac{\alpha_k}{1}$$

en correleert de stabiliteit aan de convergentie.

. Voor een overzicht van de technieken bij het approximeren van functies zie: de syllabus van de werkgroep approximatie van functies TEMME & V.D. LAAN [51], het boek van HART c.s. [20] of het overzichtsverhaal van GAUTSCHI [15] en de aldaar gegeven referenties.

. Als wij een nauwkeurige implementatie met een goede complexiteit hebben verkregen dan blijkt dat tegenwoordig hogere eisen worden gesteld t.a.v. de aflevering van het resultaat: behalve de getalwaarde wordt informatie over de nauwkeurigheid verwacht. Dit laatste zal wederom enig rekenwerk vergen, waarbij het streven is: beduidend minder - een orde lager - dan het rekenwerk nodig voor het resultaat.

3. MATRIX VERMENIGVULDIGING

In de numerieke lineaire algebra is matrix vermenigvuldiging en speciaal vermenigvuldiging met zogenaamde elementaire matrices van groot belang. Elementaire orthogonale matrices zijn

$$\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \quad \text{de vlakke rotatie;}$$

en

$$I - 2ww^H \quad \text{met} \quad w^H w = 1, \quad \text{de Householder reflectie.}$$

Een elementaire eliminatie matrix is

$$\begin{pmatrix} 1 & 0\ldots0 \\ \hline m_1 & \\ m_2 & I_k \\ \vdots & \\ m_k & \end{pmatrix} .$$

Verder zouden wij de orthogonale matrix behorende bij de Discrete Fourier Transformatie als een elementaire matrix willen bestempelen, die o.a. gebruikt wordt om speciale matrix-maal-vector producten te vereenvoudigen en niet zozeer om "nullen" te introduceren zoals in een eliminatie procedure.

3.1. Het eigenwaarden- en eigenvectorenprobleem

Een algemene methode is de zogenaamde QR-iteratie. De methode berust op het invariant zijn van de eigenwaarden onder een gelijkvormigheidstransformatie en op de zogenaamde Schurdecompositie:

$$\exists Q \quad \text{met} \quad A = QRQ^H, \qquad Q^HQ = I$$

met R bovendriehoeks.
Constructief gaat men als volgt te werk

. Reductie naar Hessenbergvorm via elementaire orthogonale matrices:

$$A = Q_0HQ_0^H, \qquad Q_0^HQ_0 = I$$

met H een Hessenberg matrix;

. Iteratie op H via elementaire orthogonale matrices

$$H_{k+1} = Q_k^H(H_k - \sigma_k I)Q_k + \sigma_k I, \qquad H_1 = H, \qquad k = 1,2,\ldots$$

waarbij onder bepaalde voorwaarden gebleken is dat $\{H_k\}$ naar een bovendriehoeksmatrix R convergeert.
Op de diagonaal van R staan benaderingen van de eigenwaarden.

OPMERKINGEN.

. A priori worden de normen van de rijen en de bijbehorende kolommen zo goed mogelijk gelijk gemaakt via een diagonaal-gelijkvormigheidstransformatie: het zogenaamde equilibreren. Bovendien kunnen rijen en kolommen verwisseld worden i.v.m. "al klare" deelmatrices.

. De reductie naar Hessenbergvorm en het toepassen van de shifts maken de QR-iteratie efficiënt. Bovendien wordt doorgaans het probleem in deelproblemen gesplitst als een benedendiagonaalelement verwaarloosbaar klein is, wederom uit efficiëntie overwegingen. Een speciaal geval is het afsplitsen van het gedeelte wat al klaar is.

. Als de matrix symmetrisch is dan is de Hessenbergvorm een tridiagonale matrix.

. De bepaling van de eigenvectoren is teruggebracht tot de bepaling van de eigenvectoren van R.
Immers, stel V is gevonden met

$$RV = V\Lambda,$$

waarbij Λ een diagonaalmatrix is met de eigenwaarden, dan geldt

$$A = Q_0Q_1\ldots Q_n V\Lambda V^{-1}(Q_0Q_1\ldots Q_n)^H.$$

De kolommen van $(Q_0Q_1\ldots Q_nV)$ vormen de eigenvectoren van A.

. In de praktijk wordt voor reële matrices geïtereerd naar blokbovendriehoeksvorm om complexe arithmetiek te vermijden; de complex geconjugeerde eigenwaarden worden gegeven in disjuncte 2-bij-2 blokjes langs de diagonaal.

. Het bepalen van de eigenwaarden via de nulpuntsbepaling van het karakteristieke polynoom, is in het algemeen een instabiel algoritme: uitgaande van de matrix moeten de coëfficiënten van het karakteristieke polynoom als tussenfase vermeden worden, als de nulpunten onacceptabel gevoelig zijn voor verstoringen in de coëfficiënten.

VOORBEELD (BAUER [4]).
Het symmetrische eigenwaardenprobleem is goed geconditioneerd.
Zij de matrix gegeven door

$$\begin{pmatrix} a & b \\ b & d \end{pmatrix}$$

dan geeft de vorming van de karakteristieke vergelijking

$$\lambda^2 - 2p\lambda + q = 0, \quad p = (a+d)/2, \quad q = ad - b^2$$

een slecht geconditioneerd deelprobleem als $p^2 \sim q \neq 0$, (samenvallende wortels), hetgeen overeenkomt met $a \sim d$, $b \sim 0$. Een stabiel algoritme is de *Jacobi-iteratie*, met als eigenwaarden $\frac{1}{2}(a+d) \pm \sqrt{((a-d)/2)^2 + b^2}$. (Deze formule is eenvoudiger af te leiden door eerst een shift, $\sigma = \frac{1}{2}(a+d)$, toe te passen en dan de resulterende karakteristieke vergelijking op te lossen.)

3.2. Fast Givens

Het vermenigvuldigen met een elementaire rotatiematrix kan vereenvoudigd worden als de matrix die vermenigvuldigd moet worden een diagonaalmatrix als factor heeft (Een willekeurige matrix is gefactoriseerd te denken in eenheidsmatrix-maal-matrix.)

Het probleem, in zijn eenvoudigste vorm, luidt:
vind $w_1', w_2', \alpha, \beta$ bij gegeven a_1, a_2, w_1, w_2 zodanig dat geldt

$$\begin{pmatrix} \sqrt{w_1'} & \\ & \sqrt{w_2'} \end{pmatrix} \begin{pmatrix} 1 & \alpha \\ \beta & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = U \begin{pmatrix} \sqrt{w_1} & \\ & \sqrt{w_2} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} v \\ 0 \end{pmatrix}$$

waarbij U orthogonaal is en zodanig gekozen dat het rechterlid evenredig aan een eenheidsvector is, en a_1, a_2 elementen zijn van een kolom van een matrix A. De oplossing hiervan is

$$\beta = -a_2/a_1, \quad \alpha = -\beta w_2/w_1$$
$$w_1' = w_1/(1-\alpha\beta)$$
$$w_2' = w_2/(1-\alpha\beta)$$

waarbij de rijen verwisseld worden als $a_1 = 0$ of $1 - \alpha\beta > 2$ om overflow respectievelijk underflow te vermijden.

OPMERKINGEN.

1. Bij de normale vlakke rotatie is een worteltrekking nodig bij de bepaling van de cosinus en de sinus; hier kan de worteltrekking uitgesteld worden en (eventueel) op het laatst gebeuren; op grond hiervan noemde GENTLEMAN het ook wel de wortelvrije Givensrotatie.
2. Deze rotaties zijn handig bij zogenaamde ijle matrices, waarbij gebruik gemaakt kan worden van de "nullen", en bij het "updaten" van een oplossing als er rijen toegevoegd worden. Dit laatste is van belang bij kleinste-kwadratenproblemen waarbij het aantal rijen niet allemaal tegelijk in het (directe) geheugen kunnen.

3. De winst in het aantal operaties t.a.v. de gewone Givensrotatie zit in de vermenigvuldiging van de 'rotatiematrix' met de overige kolommen van A.
4. Andere toepassingen van 'updating van factorisaties' zijn gegeven door GILL c.s. [17,18] en BUNCH c.s. [6]. PAIGE [37] is nader ingegaan op de foutenanalyse van enige orthogonale factorisaties.

3.3. Een-dimensionale Discrete Fourier Transformatie

De DFT is het matrix-maal-vector product

$$W_n v = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & w_n & & w_n^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & w_n^{n-1} & & w_n^{(n-1)^2} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}, \qquad w_n = \exp(2\pi i/n).$$

Technieken om dit snel te doen staan bekend onder de naam FFT (Fast Fourier Transform).
Als de orde van de matrix factoriseerbaar is, dan kunnen de *Cooley-Tukey-achtige* algoritmen (zie 3.3.1) toegepast worden. Als de orde van de matrix een priemgetal p is, dan kan de *Rader-permutatie* (zie 3.3.2) gebruikt worden om het probleem terug te voeren op een circulaire correlatie van de orde p - 1. Deze circulaire correlaties kunnen dan wederom in DFT's uitgedrukt worden (zie 3.4), of via WINOGRAD's techniek uitgerekend worden als p - 1 te factoriseren is in onderlinge priemgetallen of machten van priemgetallen (zie 3.3.3).
Programmatuur m.b.t. de diverse DFT-implementaties en toepassingen is verschenen in WEINSTEIN c.s. [61].

3.3.1. De Cooley-Tukey-achtige DFT

De berekening van

$$A_k = \sum_{\ell=0}^{n-1} a_\ell w_n^{k\ell}, \qquad k = 0,1,\ldots,n-1, \qquad w_n = \exp(2\pi i/n)$$

wordt gereduceerd tot kleinere problemen als geldt $n = n_1 \times n_2$.
Als wij schrijven

$$k = k_1 n_2 + k_2, \qquad 0 \le k_2 < n_2, \quad 0 \le k_1 < n_1$$
$$\ell = \ell_1 n_1 + \ell_2, \qquad 0 \le \ell_2 < n_1, \quad 0 \le \ell_1 < n_2$$

dan geldt bijvoorbeeld

$$A_{k_1 k_2} = \sum_{\ell_2=0}^{n_1-1} \left(W_n^{k_2 \ell_2} \left(\sum_{\ell_1=0}^{n_2-1} a_{\ell_1 \ell_2} W_{n_2}^{k_2 \ell_1} \right) W_{n_1}^{k_1 \ell_2} \right),$$

waarbij de binnensom staat voor n_1 DFT's van de orde n_2, en de buitensom staat voor n_2 DFT's van de orde n_1; de factoren $W_n^{k_2 \ell_2}$ heten draaifactoren. Als wij de DFT's van orde n_1 en n_2 berekenen via polynoomevaluatie en de vorming van en vermenigvuldiging met de draaifactoren verwaarlozen, dan is de hoeveelheid werk gelijk aan

$$n_1 n_2^2 + n_2 n_1^2 = n(n_1 + n_2),$$

i.p.v. n^2 voor het 'grote' probleem.
Dit proces kan herhaald worden door de factoren verder te factoriseren. Varianten van deze algoritme zijn gebaseerd op andere representaties van k en ℓ en op diverse technieken t.a.v. het geheugenbeheer. Vermeldenswaard is de Thomas/Good-algoritme genoemd in COOLEY c.s. [10], waarbij de representaties van k en ℓ luiden

$$k = n_1 k_1 + k_2 n_2 \pmod{n}, \quad 0 \le k_1 < n_2, \; 0 \le k_2 < n_1$$
$$\ell = n_2 p_2 \ell_1 + n_1 p_1 \ell_2 \pmod{n}$$

met n_1 en n_2 relative priemgetallen en

$$n_2 p_2 = 1 \pmod{n_1}, \qquad n_1 p_1 = 1 \pmod{n_2}.$$

Het voordeel is dat er geen draaifactoren optreden, omdat

$$W_n^{k\ell} = W_{n_1}^{k_2 \ell_1} W_{n_2}^{k_1 \ell_2}.$$

3.3.2. De Rader-permutatie

RADER [40] heeft opgemerkt dat de DFT van orde p, met p een priemgetal, als essentieel deelprobleem een circulaire correlatie bevat. Dit berust op

de eigenschap dat {1,...,p-1} op zichzelf afgebeeld kan worden via

$$\exists g \in \mathbb{N} \text{ met } i \mapsto g^i \pmod p, \quad i = 1,2,\ldots,p-1.$$

Als we de DFT schrijven als

$$A_\ell - a_0 = \sum_{k=1}^{p-1} a_k w_p^{k\ell}$$

en vervolgens permuteren door te substitueren

$$\ell = ((g^\ell)), \quad k = ((g^k))$$

(de dubbele haakjes duiden op modulo-p-rekenen) dan verkrijgen wij de circulaire correlatie:

$$A_{((g^\ell))} - a_0 = \sum_{k=1}^{p-1} a_{((g^k))} w_p^{(g^{k+\ell})}.$$

Als wij nu deze correlaties in DFT's uitdrukken dan is hiermee de complexiteit verkleind.

3.3.3. De een-dimensionale Winograd DFT

Winograd beschouwt DFT's met als orden: een priemgetal, een macht van een priemgetal (is niet uitgewerkt in zijn publicatie [60]) of een product van getallen die relatief priem zijn.

DFT van orde priemgetal

WINOGRAD [59,60] factoriseert de DFT-matrix, van de orde een priemgetal, in de vorm

SCT

met

S,T, incidence matrices (bevatten alleen 0, ± 1 als elementen),
C een diagonaalmatrix met op de diagonaal óf zuiver reële of zuiver imaginaire elementen.

VOORBEELDEN

$$W_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & w & w^2 \\ 1 & w^2 & w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & & \\ & \cos u - 1 & \\ & & i \sin u \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix}, \quad u = 2\pi/3,\ w = e^{iu}$$

$$W_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & i \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} .$$

Door deze factorisatie komt Winograd op minder vermenigvuldigingen, $O(p)$, - ze zijn ook nog eenvoudiger! - dan de directe algoritmen, $O(p^2)$ - evaluatie van het polynoom voor equidistante argumenten - terwijl het aantal optellingen, bij een geschikt gekozen *volgorde* in de uitwerking van incidence-matrix-maal-vector, ruwweg gelijk blijft.

Winograd's techniek is:

. Bij een DFT van de orde priem, p, is de operatie DFT-matrix-maal-vector na permutatie te interpeteren als een cyclische correlatie (RADER [40]);

. De cyclische correlaties kunnen opgevat worden als een product van polynomen, in de variabele y zeg, modulo y^p-1.
Dit product kan teruggebracht worden tot polynoomproducten modulo polynomen van lagere graad. Deze laatste bewerkingen kunnen snel.

STELLING (WINOGRAD [60]).
Zij R_ℓ *en* S_m *polynomen van de graad* ℓ *en* m, *en zij* $P = P_1P_2$ *een polynoom van de graad* n *met graad* $(P_1) = n_1$, *graad* $(P_2) = n_2$, P_1 *en* P_2 *relatief priem, dan geldt*

$$P_\ell S_m \pmod P = (Q_2P_2(R_\ell S_m \pmod{P_1}) + Q_1P_1(R_\ell S_m \pmod{P_2})) \pmod P$$

met Q_1 *en* Q_2 *polynomen zo, dat*

$$Q_1P_1 + Q_2P_2 = 1 \pmod P.$$

ILLUSTRATIE (factorisatie W_3)
Als deelprobleem in W_3-maal-vector hebben wij

$$\begin{pmatrix} w & w^2 \\ w^2 & w \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix}, \quad \text{met } w = \exp(iu) \text{ en } u = 2\pi/3.$$

Deze cyclische correlatie is equivalent aan het berekenen van de coëfficiënten van het product van de twee polynomen in y

$$(w+w^2y)(x_2+x_3y) \quad (\mathrm{mod}(y^2-1)).$$

Via de bovenstaande stelling, waarbij $Q_1 = -Q_2 = \frac{1}{2}$ is gekozen, reduceert dit tot

$$\{\tfrac{1}{2}(y+1)((w+w^2y)(x_2+x_3y) \ (\mathrm{mod}(y-1))$$
$$-\tfrac{1}{2}(y-1)((w+w^2y)(x_2+x_3y) \ (\mathrm{mod}(y+1))\} \ (\mathrm{mod}(y^2-1))$$

met als resultaat

$$\tfrac{1}{2}\{(w+w^2)(x_2+x_3)+(w-w^2)(x_2-x_3) + ((w+w^2)(x_2+x_3) - (w-w^2)(x_2-x_3))y\}.$$

Bij substitutie van

$$\tfrac{1}{2}(w+w^2) = \cos 2\pi/3$$
$$\tfrac{1}{2}(w-w^2) = i \sin 2\pi/3$$

verkrijgen wij

$$\begin{pmatrix} w & w^2 \\ w^2 & w \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \cos u & \\ & i \sin u \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \end{pmatrix},$$

en tenslotte

$$W_3 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & & \\ & \cos u - 1 & \\ & & i \sin u \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix}.$$

De factorisatie van W_p, met $p-1 = p_1p_2$ en p_1 en p_2 relatieve priemgetallen, herleidt Winograd tot een nesting van circulaire correlaties.

ILLUSTRATIE (factorisatie W_7)

$$\begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ \vdots \\ A_6 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & \dots & w^6 \\ 1 & w^2 & w^4 & & \vdots \\ \vdots & \vdots & & & \vdots \\ 1 & w^6 & \dots & & w^{36} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_6 \end{pmatrix} \quad , \quad w = \exp(2\pi i/7)$$

Na een algemenere permutatie dan die van Rader waarbij nu een blokcirculant ontstaat, hebben wij

$$\left(\begin{array}{c} A_0 \\ \cdots \\ \Phi_0 \\ \cdots \\ \Phi_1 \end{array}\right) = \left(\begin{array}{c:ccc:ccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hdashline 1 & & & & & & \\ 1 & & X_0 & & & X_1 & \\ 1 & & & & & & \\ \hdashline 1 & & & & & & \\ 1 & & X_1 & & & X_0 & \\ 1 & & & & & & \end{array}\right) \left(\begin{array}{c} a_0 \\ \cdots \\ Y_0 \\ \cdots \\ Y_1 \end{array}\right)$$

met

$$\Phi_0 = \begin{pmatrix} A_1 \\ A_4 \\ A_2 \end{pmatrix} \quad , \quad \Phi_1 = \begin{pmatrix} A_6 \\ A_3 \\ A_5 \end{pmatrix}$$

$$X_0 = \begin{pmatrix} w & w^4 & w^2 \\ w^4 & w^2 & w \\ w^2 & w & w^4 \end{pmatrix}, \quad X_1 = \begin{pmatrix} w^6 & w^3 & w^5 \\ w^3 & w^5 & w^6 \\ w^5 & w^6 & w^3 \end{pmatrix}$$

$$Y_0 = \begin{pmatrix} a_1 \\ a_4 \\ a_2 \end{pmatrix} \quad , \quad Y_1 = \begin{pmatrix} a_6 \\ a_3 \\ a_5 \end{pmatrix} \quad .$$

Toepassing van de resultaten verkregen bij cyclische correlatie van orde 2 geeft:

$$\begin{pmatrix} \Phi_0 \\ \Phi_1 \end{pmatrix} = \begin{pmatrix} I & I \\ I & -I \end{pmatrix} \begin{pmatrix} \frac{1}{2}(X_0+X_1) & \\ & \frac{1}{2}(X_0-X_1) \end{pmatrix} \begin{pmatrix} I & I \\ I & -I \end{pmatrix} \begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix}$$

(N.b. er zijn nog maar twee circulant-maal-vector operaties van de orde 3 nodig!)

Circulant-maal-vector van de orde 3 kan berekend worden door gebruik te maken van de factorisatie

$$\begin{pmatrix} x_0 & x_1 & x_2 \\ x_1 & x_2 & x_3 \\ x_2 & x_3 & x_1 \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 1 & 1 & -2 & 1 \\ 1 & 1 & 1 & -2 \\ 1 & -2 & 1 & 1 \end{pmatrix}\begin{pmatrix} (x_0+x_1+x_3) & & & \\ & (x_0-x_2) & & \\ & & (x_1-x_2) & \\ & & & (x_0-x_1) \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \\ 1 & 0 & -1 \end{pmatrix}.$$

WINOGRAD [60] geeft circulaire correlaties van de orden 2,3,4,5,6, DFT's van de orden 2,3,4,5,7,8,9,16; de DFT van de orde 12 is behandeld als speciaal geval van de orden n_1n_2, met n_1 en n_2 relatief priem en gegeven factorisatie van W_{n_1} en W_{n_2}. Bovendien stipt hij de mogelijkheid aan waarbij de orde een macht van een priemgetal is.

SILVERMAN [47] is nader ingegaan op de DFT van de orde $n = n_1n_2..n_k$, met alle n_i relatief priem. Voor $n = n_1n_2$ onderkent hij

$$W_n = P_e(W_{n_1} * W_{n_2})P_b$$

waarbij * het Kroneckerproduct betekent en P_e,P_b permutatiematrices zijn. Door bovendien de Winograd-factorisatie voor de kleine W matrices te gebruiken

$$W = SCT$$

én te onderkennen dat geldt (zie MARCUS & MING [31]) voor willekeurig A,B, C,D

$$AB*CD = (A*C)(B*D)$$

verkrijgt hij

$$W_{n_1n_2} = P_e(S_{n_1\ell_1} * S_{n_2p_2})(C_{\ell_1\ell_1} * C_{\ell_2\ell_2})(T_{\ell_1n_1} * T_{\ell_1n_2})P_b.$$

Uitbreiding van het bovenstaande geeft de factorisatie voor de orde

$n = n_1 n_2 \ldots n_k$ met de n_i relatief priem.

3.3.4. De conditie van de DFT

Als conditie van de DFT hebben wij als speciaal geval van §1.1.:

$$C(v) = \frac{\|W(v+\Delta v)-Wv\|_2}{\|Wv\|_2} \Big/ \frac{\|\Delta v\|_2}{\|v\|_2} = 1.$$

3.3.5. De groeifactoren van de diverse algoritmen

Het idee is dat voor een algoritme a geldt: $\exists \Delta v_a$ met

$$fl_a(Wv) \equiv W(v+\Delta v_a)$$

met

$$\frac{\|\Delta v_a\|_2}{\|v\|_2} = g_a * \varepsilon.$$

Een lijstje van groeifactoren is niet gepubliceerd voor de diverse snelle DFT's.

3.4. Discrete convolutie

De convolutie is gedefinieerd door

$$\sum_{i=0}^{n-1} c_{j-i} b_i, \qquad j = 0,1,\ldots,n-1.$$

In matrixnotatie is de convolutie het volgende matrix-maal-vector product:

$$\begin{pmatrix} c_0 & c_{-1} & & c_{-(n-1)} \\ c_1 & c_0 & \ddots & \vdots \\ c_2 & c_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & c_{-1} \\ c_{n-1} & \cdots\cdots & c_1 & c_0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{pmatrix}.$$

Een matrix waarvoor geldt dat de elementen een functie zijn van het verschil van de indices heet een Toeplitz-matrix. Het bovenstaande probleem luidt dus: Toeplitz-matrix-maal-vector.
Een bijzonder geval van bovenstaand probleem is de circulaire convolutie;

de matrix wordt dan een circulant genoemd (c is periodiek, d.w.z. $c_{-k} = c_{n-k}$, $k = 1,2,\ldots,n-1$).

In het onderstaande worden enige lemma's en theorema's gegeven m.b.t. circulanten en Toeplitz-matrices. Voor de bewijzen wordt verwezen naar VAN DER LAAN [56]. Verder richten wij ons op Toeplitz-matrices; Hankel-matrices kunnen door spiegeling overgevoerd worden in Toeplitz-matrices en worden hier derhalve verder niet meer genoemd. Correlaties en convoluties hangen samen doordat correlaties als Hankel-matrix-maal-vector gezien kunnen worden.

Een andere benadering voor speciale orden n is gegeven door WINOGRAD [59,60] i.v.m. zijn DFT. Verder zijn er nog allerlei transformaties, bijvoorbeeld de 'number theoretic DFT' die dit matrix-maal-vector product vereenvoudigen. Wij gaan daar niet verder op in.

LEMMA 3.1 (Eigensysteem van een circulant)
Een circulant C(c) *kan gefactoriseerd worden als*

$$C(c) = n^{-1} W \Lambda(c) \bar{W},$$

met

$$W = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & w & & w^{n-1} \\ \vdots & & \ddots & \vdots \\ 1 & w^{n-1} & \ldots & w^{(n-1)^2} \end{pmatrix}, \quad w = \exp(2\pi i/n)$$

$$\Lambda(c)_{ij} = \begin{cases} (\bar{W}c)_j & , \quad i = j \\ 0 & , \quad i \neq j \end{cases}.$$

THEOREMA 3.1 (Circulant-maal-vector)

$$C(c)b = n^{-1} W(\Lambda(c) \quad (\bar{W}b)).$$

De hoeveelheid rekenwerk is 3-maal een DFT, dus $3n \sum_i p_i$ met $n = \prod_i p_i$.

(Als n priem is dan is de DFT toch snel uit te voeren, zie 3.3.)

LEMMA 3.2 (Representatie van een bovendriehoeks-Toeplitz-matrix als een som van een circulant en een diagonaal-gelijkvormigheidstransformatie van een circulant.)

$$\nabla(c) = \tfrac{1}{2}\{C(c) + D'C(c')\bar{D}'\}$$

met

$\nabla(c)$ *is bovendriehoeksmatrix van* $C(c)$,

$$c' = \begin{pmatrix} 1 & & \\ & -1 & \\ & & -1 \end{pmatrix} \bar{D}'c,$$

$$D' = \begin{pmatrix} 1 & & \\ & v & \\ & & v^{n-1} \end{pmatrix}, \quad v = \exp(\pi i/n).$$

THEOREMA 3.2 (Driehoeks-Toeplitz-matrix-maal-vector)

$$\nabla(c)b = \{W\,\Lambda(c)\,\bar{W}b + D'\,W\,\Lambda(c')\,\bar{W}\,\bar{D}'b\}/(2n).$$

OPMERKING. De coëfficiënten van het (Cauchy) product van twee polynomen, zowel in de machtssomrepresentatie als som van Chebyshevpolynomen, kan herleid worden tot driehoeks-Toeplitz-matrix-maal-vector operaties; deze operaties kunnen efficiënt uitgevoerd worden via de DFT.

LEMMA 3.3 (Representatie van een Toeplitz-matrix als een verschil van een circulant en een diagonaal-gelijkvormigheidstransformatie van een circulant.)

$$T(c) = C(\tfrac{1}{2}(c+E^{-}Sc_{-})) - D'\,C(\tfrac{1}{2}(c-E^{-}Sc_{-}))\,\bar{D}'$$

met

$$c_{-} = (c_0, c_{-1}, \ldots, c_{-(n-1)})^T$$

$$S = \begin{pmatrix} 0 & & & 1 \\ & & \iddots & \\ & 1 & & \\ 1 & & & 0 \end{pmatrix}$$

$$E^{-} = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 1 & 0 & & 0 \\ & 1 & \ddots & \\ & & 1 & 0 \end{pmatrix}$$

en voor D' *zie Lemma* 3.2.

THEOREMA 3.3 (Toeplitz-matrix-maal-vector)

$$T(c)\, b = W \Lambda (c+E^{-}Sc_{\_})\, \bar{W} b - D'\, W \Lambda ((c-E^{-}Sc_{\_})')\, \bar{W}\, \bar{D}' c\, /(2n).$$

OPMERKING. In bovengenoemde theorema's zijn vierkante matrices beschouwd; door de vector b met nullen aan te vullen kunnen wij het bovenstaande toepassen op een rechthoekige matrix met meer rijen dan kolommen.

4. LINEAIRE STELSELS

Bij het oplossen van lineaire stelsels wordt onderscheid gemaakt tussen directe en iteratieve methoden. Bij directe methoden wordt de matrix gefactoriseerd, terwijl bij iteratieve methoden een splitsing t.a.v. de optelling van belang is. In het onderstaande willen wij ons beperken tot enige directe methoden.
Zij x te bepalen uit

$$Ax = b$$

in eindige precisie arithmetiek.
In zijn algemeenheid geldt dat de verkregen oplossing, $x + \Delta x$, exact voldoet aan

$$(A+\Delta A)(x+\Delta x) = b + \Delta b$$

waarbij

$$\Delta A = \Delta A_m + \Delta A_a$$

met A_m, Δb de verstoringen t.g.v. meetfouten of conversiefouten en ΔA_a het effect van de methode in de eindige arithmetiek. Als grens voor de fout kunnen wij a priori verkrijgen

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\|A^{-1}\|\|A\|}{1-\|A^{-1}\Delta A\|} \left\{\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A_m\|}{\|A\|} + g*\varepsilon\right\}$$

onder de aanname $\|A^{-1}\Delta A\| \ll 1$, waarbij ε de precisie is van de (eindige) arithmetiek en g de groeifactor van de methode.

Een schatting van de fout is a posteriori te verkrijgen door $x + \Delta x$ te beschouwen als oplossing van

$$A(x+\Delta x) = b + \Delta b + \Delta b_a$$

met $\Delta b_a = r$, de residuvector. Er geldt dan

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\|\|A^{-1}\| \left\{\frac{\|\Delta b\|}{\|b\|} + \frac{\|r\|}{\|b\|}\right\} ;$$

STEWART [49] vermeldt dat dergelijke grenzen veelal niet scherp zijn omdat

$$\|b\| \leq \|A\|\|x\|$$

veelal niet scherp is.

FORSYTHE & MOLER [12] geven als grens voor ΔA_a voor de algoritme voor Gauss eliminatie met pivoten

$$\|\Delta A_a\|_\infty \leq 1.01(n^3+3n^2)g\|A\|_\infty \varepsilon$$

waarbij voor partieel pivoten geldt $g \leq 2^{n-1}$ en voor compleet pivoten geldt $g \leq 1.8n^{.25\ln(n)}$.

In tegenstelling tot de handelwijze bij approximatie van functies kunken wij gezien het grote aantal mogelijkheden niet allerlei matrices systematisch verifiëren en zo een nauwkeurigheidsprofiel verkrijgen; wij moeten ons laten leiden door de grenzen van de fout.

De complete pivotstrategie levert naast de eliminatie nog eens het

zoeken van de pivot op dat in zijn totaliteit een $O(n^3)$ proces is. Bovendien weet men op grond van experimenten dat alhoewel de groeifactor, 2^{n-1}, gehaald kan worden, dit in matrices die in de praktijk voorkomen niet gebeurt. Een oplossing van dit dilemma zou kunnen zijn gemengd te pivoten.

De eliminatie-algoritmen berusten op

. decompositie van A = LR (of QR) al dan niet volledig;

. voorwaartse substitutie

$$L(Rx) = b \rightarrow Rx = L\backslash b;$$

. terugwaartse substitutie

$$x = R\backslash L\backslash b.$$

Bij een symmetrisch positief definiet stelsel kunnen wij goedkoper de factorisatie verkrijgen in de vorm

$$A = L\ D\ L^T.$$

Jaren lang was er voor het symmetrische indefiniete stelsel geen numeriek stabiel algoritme; een overzicht van de huidige stabiele technieken is gegeven door BARWELL & GEORGE [3].

4.1. Speciale lineaire stelsels

In deze paragraaf geven wij enige theorema's; voor bewijzen en bronvermeldingen verwijzen wij naar VAN DER LAAN [56].

THEOREMA 4.1 (Oplossen van een stelsel waarvan de matrix een circulant is)

$$C(c)\ x = b \Rightarrow x = n^{-1}W(\ \Lambda^{-1}(c)\ (\bar{W}b))$$

onder de aanname $\Lambda(c)$ *is niet singulier.*

THEOREMA 4.2 (Oplossen van een stelsel waarvan de matrix een driehoeks-Toeplitz-matrix is)

$$\nabla(c)\ x = b \Leftrightarrow \begin{pmatrix} \nabla & T \\ & \nabla \end{pmatrix} \begin{pmatrix} x_1 \\ \hline x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ \hline b_2 \end{pmatrix}$$

en kan dus recursief worden teruggebracht tot het oplossen van kleinere stelsels van dezelfde structuur; de benodigde Toeplitzmatrix-maal-vector producten kunnen snel uitgevoerd worden.

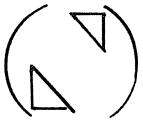
THEOREMA 4.3 (Oplossen van een stelsel waarvan de matrix een band-Toeplitz-matrix is)
Zij

$$Tx = b$$

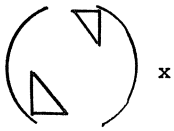
met T *band-Toeplitz met* ℓ *beneden codiagonalen en* u *boven codiagonalen, dan geldt*

$$(C - \text{  }) \; x = b$$

en voor reguliere C

$$x - C^{-1}\,\text{  }\; x = C^{-1}b.$$

Als wij stellen

$$\hat{x} = \text{  }\; x$$

dan is het stelsel gereduceerd tot het $(\ell+u)$*-stelsel*

$$x_k - (C^{-1}\hat{x})_k = (C^{-1}b)_k, \qquad k = 1,\ldots,u \text{ en } n-\ell+1,\ldots,n;$$

de resulterende onbekenden worden expliciet gegeven door

$$x_k = (C^{-1}(b+\hat{x}))_k, \qquad k = u+1,\ldots,n-\ell.$$

De laatste jaren is er nogal wat aandacht besteed aan de "snelle Poisson oplossers". HENRICI [22] interpreteert discretisaties van de

Poisson-vergelijking als meer-dimensionale convoluties, binnen een ruimte van periodieke vectoren met periode 2n.

Zij de Poissonvergelijking gegeven door

$$\Delta u = f$$

en een discretisatie door

$$D*U = h^2 F$$

dan geldt voor de oplossing

$$U = \frac{h^2}{4n^2} \left(F_{2n}^{(2)} \right)^{-1} \left\{ \left(F_{2n}^{(2)} F \right) / F_{2n}^{(2)} D \right\}$$

waarbij de deling elementsgewijs opgevat moet worden en $F_{2n}^{(2)}$ de 2-dimensionale DFT is.

OPMERKING. In SZMYDT [50] wordt een analogon voor lineaire (partiële) differentiaal operators met constante coëfficiënten toegepast; i.p.v. eerst te discretiseren worden de (continue) transformaties toegepast, pas later tijdens het concreet uitwerken van de integraaltransformaties wordt er gediscretiseerd.

4.2. Speciale lineaire kleinste-kwadratenproblemen

Bij een overbepaald stelsel waarbij de matrix Toeplitz is, kan men efficiënt een oplossing verkrijgen door:

- de matrix naar bi-diagonaalvorm te transformeren via de Lanczos-algoritme (de Toeplitz-matrix-maal-vector operaties kunnen snel, zie 3.4)
- het resulterende bi-diagonaal stelsel oplossen, eventueel met regularisatie.

Zie: Dianne P. O'Leary, J.A. Simmons, A bidiagonalization-regularization procedure for large scale descretizations of ill-posed problems. Techn. Rep. (1980) Univ. of Maryland.

5. LITERATUUR

[1] AHO, A.V., H.E. HOPCROFT & J.D. ULLMAN, *The design and analysis of computer algorithms*, Addison Wesley, 1974.

[2] BABUSKA, I., *Numerical stability in problems of linear algebra*, SIAM J. Numer. Anal, 9 (1972) 53-77.

[3] BARWELL, V., A. GEORGE, *A comparison of algorithms for solving symmetric indefinite systems of linear equations*, TOMS, 2 (1976) 242-251.

[4] BAUER, F.L., *Computational graphs and rounding error*, SIAM J. Numer. Anal, 11 (1974) 87-96.

[5] BORODIN, A., I. MUNRO, *The computational complexity of algebraic and numeric problems*, Elsevier, 1975.

[6] BUNCH, J.R., C.P. NIELSEN, *Updating the singular value decomposition*, Numer. Math. (1978) 111-129.

[7] CODY, W.J., H.C. THACHER Jr., *Rational Chebyshev approximations for the exponential integral* $E_1(x)$, Math. Comp., 22 (1968) 641-649.

[8] CODY, W.J., H.C. THACHER Jr., *Chebyshev approximations for the exponential integral* $E_i(x)$, Math. Comp., (1969) 289-303.

[9] CODY, W.J., *Another aspect of economical polynomials*. Letters to the editor, Comm. ACM (1967) 537.

[10] COOLEY, J.W., P.A.W. LEWIS, P.D. WELCH, *Historical notes on the fast Fourier transform*. In: RABINER c.s. [39], 260-262.

[11] FIKE, C.T., *Methods of evaluating polynomial approximations in Function evaluation routines, Comm. ACM (1967) 175-178.*

[12] FORSYTHE, G.E., C.B. MOLER, *Computer solution of linear algebraic systems*, Prentice Hall, 1967.

[13] FORSYTHE, G.E., M.A. MALCOLM, C.B. MOLER, *Computer methods for mathematical computations*, Prentice Hall, 1977.

[14] FORSYTHE, G.E. *Pitfalls in computation, or why a math book isn't enough*, Amer. Math. Monthly, 77 (1970) 931-956.

[15] GAUTSCHI, W., *Computational methods in special functions - A survey.* In: ASKEY, R., *Theory and applications of special functions,* 1975, Proceedings seminar Univ. Wisconsin.

[16] GENTLEMAN, W.M., *Matrix multiplication and Fast Fourier Transforms,* Bell System Techn. J. (1968) 1099-1103.

[17] GILL, P.E., G. H. GOLUB, W. MURRAY, M.A. SAUNDERS, *Methods for modifying matrix factorizations,* Math. Comp (1974) 505-535.

[18] GILL, P.E., W. MURRAY, M.A. SAUNDERS, *Methods for computing and modifying the LDV factors of a matrix,* Math. Comp (1975) 1051-1077.

[19] GINSBURG, M., *A guide to the literature of modern numerical mathematics,* Comp. Rev. (1975) 83-97.

[20] HART, J.F., E.W. CHENEY, C.L. LAWSON, H.J. MAEHLY, C.K. MESZTENYI, J.R. RICE, H.C. THACHER Jr., C. WITZGALL, *Computer Approximations,* Krieger Publishing Company.

[21] HARTMANIS, J. & J.E. HOPCROFT, *An overview of the theory of computational complexity,* J. ACM (1971) 444-475.

[22] HENRICI, P., *Fast Fourier Methods in computational complex analysis,* SIAM Rev. (1979) 481-527.

[23] HOLLENBERG, J.P., *Continued fractions,* in: N.M. TEMME, C.G. VAN DER LAAN (eds.) *Syllabus approximatie van functies,* 1981 (te verschijnen).

[24] HOPCROFT, J.E., *Complexity of computer computations,* IFIP congres (1974) 620-626.

[25] HOUSEHOLDER, A.S., *Generation of errors in digital computation,* Bull. Amer. Math. Society (1954) 234-247.

[26] LARSON, J., A. SAMEH, *Efficient calculation of the effect of roundoff errors,* TOMS, 4,3 (1978) 228-236; 5,3 (1979) 372.

[27] JACOBS, D.A.H. (ed.), *The state of the art in numerical analysis,* Academic Press 1977.

[28] KAHAN, W., *A survey of error analysis,* IFIP congres (1972) 1214-1239.

[29] KAHANER, D.K., *Matrix description of the Fast Fourier Transform,* IEEE Trans. AU (1970) 442-450.

[30] KNUTH, D.E., *The art of computer programming,* vol. 2: *Seminumerical algorithms,* Addison-Wesley Publ. Comp. Reading Mass, 1969.

[31] MARCUS, M., H. MINC, *A survey of matrix theory and matrix inequalities,* 1964, Allyn & Bacon.

[32] MILLER, W., *Software for roundoff analysis,* TOMS (1975) 108-128.

[33] MILLER, W. & D. SPOONER, *Software for roundoff analysis II,* TOMS (1978) 369-387.

[34] MILLER, R.E. & J.W. THATCHER, *Complexity of computer computations,* 1972, Plenum Press.

[35] OPPENHEIM, A.V. etc., *Digital Signal Processing II,* 1976, IEEE.

[36] POOLE, W.G., G. VOIGT, *Bibliography* 35. *Numerical algorithms for parallel and vector computers: An annotated bibliography,* Comp. Rev. (1974) 379-388.

[37] PAIGE, C.C., *Error analysis of some techniques for updating orthogonal decompositions,* Math. Comp. (1980) 465-471.

[38] RABIN, M.O., *Complexity of computations,* Comm. ACM (1977) 625-633.

[39] RABINER, L.R., C.M. RADER, *Digital Signal Processing,* 1972, IEEE.

[40] RADER, C.M., *Discrete Fourier transforms when the number of data samples is prime,* Proc. IEEE (1968) 1107-1108; also in: RABINER, L.R. c.s. [39].

[41] RAMOS, G., *Roundoff error analysis of the Fast Fourier Transform,* Math. Comp. (1971) 757-768.

[42] RICE, J.R., C.W. GEAR, J. ORTEGA, B. PARLETT, M. SCHULTZ, L.F. SHAMPINE, P. WOLFE, J.F. TRAUB, *Numerical Computation, its nature and research directions,* Signum Newsletter, February 1979.

[43] RICE, J.R., *On the condition of polynomial and rational forms,* Numer. Math. (1965) 426-435.

[44] RUTISHAUSER, H., *Zur Problematik der Nullstellenbestimmung bei Polynomen,* 1967 in: DEJON, B. (ed.), *Constructive aspects of the fundamental theorem of algebra,* John Wiley (1969) 281-294.

[45] RUTISHAUSER, H., *Endliche Arithmetik,* Kapitel 13 in: Vorlesungen über Num. Mathematik, Birkhäuser.

[46] SCHONFELDER, L., *The production of special function software in the NAG library,* Computer Centre Report, 1978, University of Birmingham.

[47] SILVERMAN, H.F., *An introduction to programming the Winograd Fourier Transform Algorithm,* IEEE Trans. Accoust. Speech and Signal Processing, ASSP (1977) 152-165.

[48] STERBENZ, P.H., *Floating point computation,* 1974, Prentice Hall.

[49] STEWART, G.W., *Introduction to matrix computations,* 1973, Academic Press.

[50] SZMIJDT, Z., *Fourier transformation and linear differential equations,* 1977, Reidel.

[51] TEMME, N.M., C.G. VAN DER LAAN, *Syllabus approximation of functions,* MC-syllabus, 1981 (te verschijnen).

[52] TRAUB, J.F. (ed.), *Analytic computational complexity,* 1976, Academic Press.

[53] TRAUB, J.F. (ed.), *Algorithms and complexity, new directions and recent results,* 1976, Academic Press.

[54] TRAUB, J.F., M. SHAW, *On the number of multiplications for the evaluation of a polynomial and some of its derivatives,* JIMA, 21 (1974) 161-167.

[55] VAN DER LAAN, C.G., *Approximatie van functies en data,* in: TE RIELE, H.J.J. (ed.), *Colloquium Numerieke Programmatuur,* MC-syllabus, 29.2 (1977) 212-279.

[56] VAN DER LAAN, C.G., *A proposal for the CO6 chapter of the NAG ALGOL 68 library,* TW 208/80, 1980, Mathematisch Centrum.

[57] WILKINSON, J.H., *Modern Error Analysis,* SIAM Rev. (1971) 548-568.

[58] WILKINSON, J.H., *Rounding errors in algebraic processes,* 1963, HMSO.

[59] WINOGRAD, S., *Arithmetic complexity of computations,* 1979, Heyden.

[60] WINOGRAD, S., *On computing the Discrete Fourier Transform,* Math. Comp. (1978) 175-199.

[61] WEINSTEIN, C.J. etc., *Programs for digital signal processing,* 1979, IEEE.

E. BESLISKUNDE

KHACHIAN'S ELLIPSOIDE-METHODE VOOR LINEAIRE PROGRAMMERING

A. SCHRIJVER

0. INLEIDING

In februari 1979 verscheen in de Proceedings van de Sovjet Akademie van Wetenschappen een artikel van L.G. Khachian [23], waarin hij liet zien dat met de zgn. "ellipsoide-methode" lineaire programmeringsproblemen kunnen worden opgelost binnen polynomiaal begrensde tijd. Deze ellipsoide-methode was eerder ontwikkeld door D.B. Judin en A.S. Nemirovskii [20,21] en N.Z. Shor [37] om willekeurige convexe programmeringsproblemen te benaderen.

Het is lange tijd een onbeantwoorde vraag geweest of LP-problemen in polynomiale tijd zijn op te lossen. Voor het bestaan van een polynomiale algorithme bestonden een aantal aanwijzingen. Zo blijkt de bekende simplex-methode voor lineaire programmering in de praktijk een zeer efficiënte algorithme (de looptijd blijkt praktisch lineair in de afmetingen van het LP-probleem), hoewel er LP-problemen zijn geconstrueerd waarvoor de simplex-methode exponentieel lange tijd vergt. Ook was het LP-probleem een van de weinige bekende problemen in de klasse $NP \cap coNP$ waarvoor nog geen polynomiale methode was gevonden (zie Garey en Johnson [13] en Van Leeuwen [28]).

Vooralsnog is de doorbraak van Khachian echter voornamelijk van theoretisch belang. De simplex-methode is efficiënt in de praktijk maar niet in theorie, terwijl Khachian's methode efficiënt is in theorie (d.w.z. polynomiaal), maar (nog) niet in de praktijk. Het polynoom dat de looptijd van Khachian's methode begrenst heeft namelijk een zeer hoge graad, en daarnaast vereisen de berekeningen een dermate hoge precisie dat de methode praktisch numeriek instabiel is. Verder onderzoek en meer ervaring met de methode zullen de praktische relevantie moeten uitwijzen, waarvoor wel een aantal essentiële verbeteringen onmisbaar lijken.

Khachian's artikel heeft een stortvloed aan publiciteit en aan onderzoek teweeg gebracht. Aanvankelijk bleef het artikel enige tijd onopgemerkt, maar na rapporten van E.L. Lawler en van P. Gács en L. Lovász [11] werd het resultaat besproken in het wetenschappelijk tijdschrift Science [25], waarna het de voorpagina's van enige kranten haalde. Vaak werden hier zeer voorbarige conclusies getrokken, zoals dat nu ieder computerprogramma versneld kan worden, dat het handelsreizigersprobleem eenvoudig op te lossen is, dat

weersvoorspellingen op langere termijn mogelijk zijn, en dat geheime codes nu sneller breekbaar zijn. De computerfabrikanten reageerden minder enthousiast; zij benadrukten terecht dat de nieuwe methode geen alternatief vormt voor de simplex-methode, en dat de bestaande software-pakketten hun waarde dus behouden, hoewel hierbij soms de valse vergelijking werd gemaakt tussen het "average case"-gedrag van de simplex-methode en het "worst case"-gedrag van de ellipsoïde-methode (vgl. McGall [31]).

Uiteraard richt een groot deel van het door Khachian's artikel aangezette onderzoek zich op de vraag hoe de nieuwe methode praktisch toepasbaar gemaakt kan worden. De totnogtoe gepubliceerde verbeteringen lijken de snelheid van de methode en haar numerieke stabiliteit echter nauwelijks te verbeteren. Daarnaast werden en worden verdere toepassingen van de methode onderzocht. Zo bleek de methode ook de polynomiale oplosbaarheid van convexe kwadratische programmering te geven (Kozlov, Tarasov en Khachian [26]), en verder tot een aantal nieuwe inzichten in de combinatorische optimalisering te leiden (Grötschel, Lovász en Schrijver [18], Padberg en Rao [35], Karp en Papadimitriou [22]). Voor een overzicht van het onderzoek naar de nieuwe methode verwijzen we naar de bibliografie van Wolfe [38] en naar het uitgebreide artikel van Bland, Goldfarb en Todd [2].

In dit artikel geven we een bespreking van de ellipsoide-methode, ingedeeld in de volgende hoofdstukken:

1. Lineaire programmering,
2. De simplex-methode,
3. Vooraf aan de ellipsoide-methode,
4. De ellipsoide-methode,
5. De vooronderstellingen,
6. De kleinste ellipsoide,
7. E_N is klein genoeg,
8. Precisie en praktische toepasbaarheid,
9. Optimaliserings- en scheidingsalgorithmen,
10. Kwadratische programmering,
11. Toepassingen in de combinatorische optimalisering,
12. Perfecte grafen en submodulaire functies,
13. Geheeltallige lineaire programmering.

1. LINEAIRE PROGRAMMERING

Een van de vele mogelijke verschijningsvormen van een lineair programmeringsprobleem (LP-probleem) is: bepaal

(1) $\quad M = \max \{cx \mid Dx \leq b\}.$

Hierin is D een m×n-matrix, b een m-vector en zijn c en x n-vectoren, en is cx het inwending product van c en x. Zonder beperking van de algemeenheid mogen we aannemen dat D, c en b geheeltallig zijn. Gevraagd wordt een algorithme om (1) te bepalen waarvan de looptijd begrensd wordt door een polynoom in de afmetingen van het LP-probleem, d.w.z. in

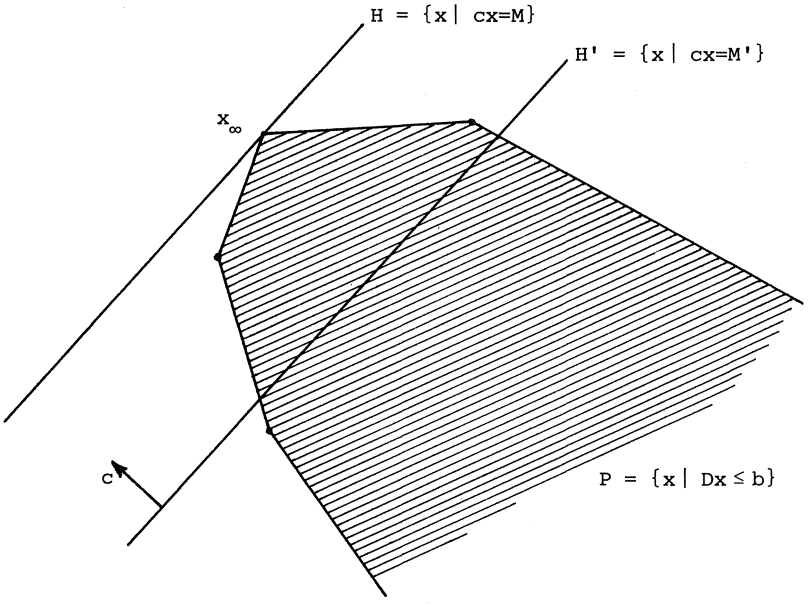
(2) $\quad (m+1)(n+1)\log T,$

waarbij T gelijk is aan het grootste getal (in absolute waarde) +1, dat voorkomt in D, c en b.

"$Dx \leq b$" representeert een stelsel lineaire ongelijkheden, en de oplossingsverzameling

(3) $\quad P := \{x \mid Dx \leq b\}$

is een polyeder in $\mathbb{Q}^n$. Meetkundig kan het LP-probleem (1) dan worden voorge-



Figuur 1.

steld als het parallel opschuiven van het hypervlak loodrecht op de vector c, net zolang als dit hypervlak nog punten van P bevat. Als x_∞ een punt van P is dat zich in het "laatste" hypervlak bevindt, dan is cx_∞ de oplossing voor (1). (Dit punt x_∞ hoeft niet uniek te zijn, noch hoeft zo'n laatste hypervlak altijd te bestaan.)

Meetkundig kan men zich eenvoudig voorstellen dat het laatste hypervlak H een niet-negative lineaire combinatie is van de facetten van P die in x_∞ samenkomen. Dit impliceert dat een vector y bestaat zo dat:

(4) $\quad y \geq 0,\ yD = c,\ yb = M.$

Dus

(5) $\quad M \geq \min \{yb \mid y \geq 0,\ yD = c\}.$

Omdat het maximum (1) niet echt groter kan zijn dan dit minimum (omdat $cx = yDx \leq yb$), volgt de Dualiteitsstelling van de lineaire programmering (Von Neumann [32], Gale Kuhn en Tucker [12]):

(6) $\quad \max \{cx \mid Dx \leq b\} = \min \{yb \mid y \geq 0,\ yD = c\}.$

Deze Dualiteitsstelling impliceert dat LP $\in$ *NP*∩co*NP*, d.w.z. dat in polynomiaal begrensde tijd *bewezen* kan worden dat het maximum (1) een zekere waarde M heeft: hiervoor hoeft men slechts een x_∞ en een y_∞ te specificeren zo dat

(7) $\quad Dx_\infty \leq b,\ y_\infty \geq 0,\ y_\infty D = c,\ cx_\infty = M = y_\infty b,$

welke berekeningen in polynomiaal-begrensde tijd kunnen worden uitgevoerd. Hoe men dit bewijs, d.w.z. zo'n x_∞ en y_∞ kan *vinden* is een moeilijker probleem; Khachian's methode toont aan dat ook dit probleem in polynomiale tijd kan worden opgelost.

2. DE SIMPLEX-METHODE

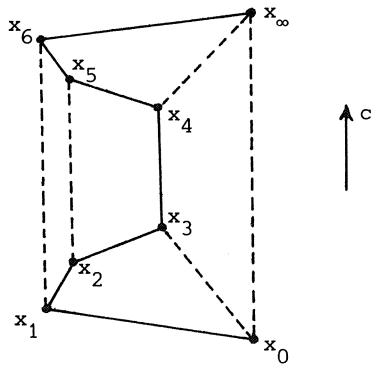
Een algorithme om het maximum (1) expliciet te bepalen is de zgn. simplex-methode, ontwikkeld door Dantzig [4]. Meetkundig gezien maakt de simplex-methode een "reis" over het polyeder P door in een hoekpunt van P te beginnen, en dan via ribben van hoekpunt naar hoekpunt te reizen, totdat

een optimaal hoekpunt is bereikt. Men kiest de te bereizen ribben zodanig dat de doelfunctie cx steeds toeneemt, althans niet afneemt.

Een meer preciese, algebraïsche uitwerking van deze ruw geschetste methode is vervat in de bekende schema's van "simplex-tableau's" en "pivot-regels", die in een eindig aantal stappen het maximum (1) opleveren. (Men moet een subroutine inbouwen om het "begin-hoekpunt" te vinden, alsmede een regel om het zgn. "cycling" te voorkomen.)

In de praktijk blijkt de simplex-methode een zeer efficiënte algorithme. Dantzig [5] rapporteert dat het aantal pivot-stappen meestal ongeveer $\frac{3}{2}$m bedraagt, en vrijwel nooit meer dan 3m. Een recent resultaat van Dantzig [6] zegt dat onder zekere voorwaarden het gemiddelde aantal pivot-stappen (over alle LP-problemen) m.log m is. Dus gemiddeld is de simplex-methode zeker polynomiaal.

De simplex-methode blijkt echter een van de weinige bekende algorithmen te zijn waarvoor het "worst case"-gedrag veel slechter is dan het "average case"-gedrag. Klee en Minty [24] vonden een klasse van LP-problemen, met m = 2n, waarvoor bij een slechte keuze van de pivot-elementen, 2^n-1 pivot-stappen moeten worden gezet. Het bijbehorende polyeder P is een kubus met schuine zijvlakken (vgl. Figuur 2 voor n = 3).



Figuur 2

Als de vector c evenwijdig loopt aan de ribbe x_0x_∞, dan kan de keuze van de pivot-stappen leiden tot een reis van x_0 naar x_∞ via de omweg x_1,x_2,x_3,x_4,x_5,x_6. Nu kan men natuurlijk ook rechtstreeks van x_0 naar x_∞ reizen, en er zijn verschillende pivot-regels opgesteld om te dwingen tot de overeenkomstige pivot-keuze (zoals "best improvement", "steepest edge"), maar ook voor deze pivot-regels zijn LP-problemen opgesteld die exponentieel veel pivot-stappen vergen (zie Jeroslow [19], Goldfarb en Sit [15]), hoofdzakelijk door de randjes van de kubus van Klee en Minty "af te schaven" zodat

afsnijpaadjes over het hoofd worden gezien.

Het is nog een open vraag of er enige pivot-regel kan bestaan die ieder LP-probleem in polynomiale tijd oplost. Hieraan verwant is het nog onopgeloste probleem of er een polynoom p(n,m) bestaat zo dat op ieder polytoop in $\mathbb{R}^n$ tussen ieder tweetal hoekpunten een pad via ribben bestaat met ten hoogste p(m,n) ribben, waarbij m het aantal facetten van P is (zie Barnette [1]).

3. VOORAF AAN DE ELLIPSOIDE-METHODE

Alvorens een beschrijving te geven van de ellipsoide-methode, enige voorbereidende opmerkingen over ellipsoïden, vooronderstellingen en hoekpunten van P.

Ellipsoïden. Een *ellipsoïde* is een verzameling vectoren van de vorm

$$(8) \qquad E = \{x \mid (x-x_0)^T A^{-1}(x-x_0) \leq 1\},$$

waarbij x_0 een vaste vector is (het *middelpunt* van E), en A een positief-definiete matrix (d.w.z. A is symmetrisch en heeft alleen positive eigenwaarden, hetgeen equivalent is met: $A = B^T B$ voor zekere nonsinguliere matrix B). De ellipsoïden zijn precies de affiene transformaties van de eenheidsbol.

Vooronderstellingen. Om de beschrijving van de ellipsoïde-methode te vereenvoudigen maken we de volgende vooronderstellingen:

(i) P is volledig-dimensionaal (d.w.z. P is niet bevat in een hypervlak);
(ii) P is begrensd;
(iii) $\max\{cx \mid x \in P\}$ wordt aangenomen in precies één hoekpunt van P.

In §5 zullen we laten zien dat deze vooronderstellingen zonder beperking der algemeenheid mogen worden gemaakt. Ieder LP-probleem kan in polynomiaal begrensde tijd worden omgezet in een LP-probleem met de eigenschappen (i), (ii) en (iii).

De hoekpunten van P. Het is eenvoudig in te zien dat ieder hoekpunt x_0 van P voldoet aan een stelsel lineaire vergelijkingen van de vorm:

(9) $D_0 x_0 = b_0$,

waarbij D_0 een niet-singuliere deelmatrix van D is, en b_0 de overeenkomstige deelvector van b. Dit volgt uit het feit dat x_0 in de doorsnede van de in x_0 samenkomende facetten van P ligt. Het stelsel vergelijkingen (9) impliceert dat de coördinaten van x_0 te schrijven zijn als quotiënten van deeldeterminanten van de matrix $[D_0\ b_0]$. Aangezien deze determinanten in absolute waarde ten hoogste

(10) $Q := n^n T^n$

zijn, weten we dat de absolute waarden van de tellers en noemers die in x_0 voorkomen eveneens ten hoogste Q zijn. Bovendien volgt

(11) $\|x_0\|^2 \leq R := n^{3n} T^{2n}$

(dit is een ruwe schatting). Merk op dat de afmetingen van Q en R (d.w.z. $\log Q$ en $\log R$) polynomiaal begrensd worden door de afmetingen van het LP-probleem.

4. DE ELLIPSOIDE-METHODE

De ellipsoïde-methode construeert een rij ellipsoïden

(12) $E_0, E_1, E_2, \ldots, E_N$,

d.w.z. rijen positief-definiete matrixen en middelpunten (vgl. (8))

(13) $A_0, A_1, A_2, \ldots, A_N$,
$x_0, x_1, x_2, \ldots, x_N$,

waarbij

(14) $N := 20\lceil n^2 . \log 12 + 13n^5 . \log n + 10n^5 . \log T \rceil$,

zo dat $x_\infty \in E_k$ voor iedere k, als volgt.

E_0 is een bol die P geheel omvat, bijvoorbeeld

(15) $E_0 = \{x \mid x^T x \leq R\}$

vanwege (11). Dus dan is $A_0 = R.I$ en $x_0 = \underline{0}$. Als $E_0, \ldots, E_k$ gedefinieerd zijn, dan wordt E_{k+1} als volgt gevonden.

I. Als $x_k \notin P$, dan is er een ongelijkheid

(16) $\quad d_i x \leq b_i$

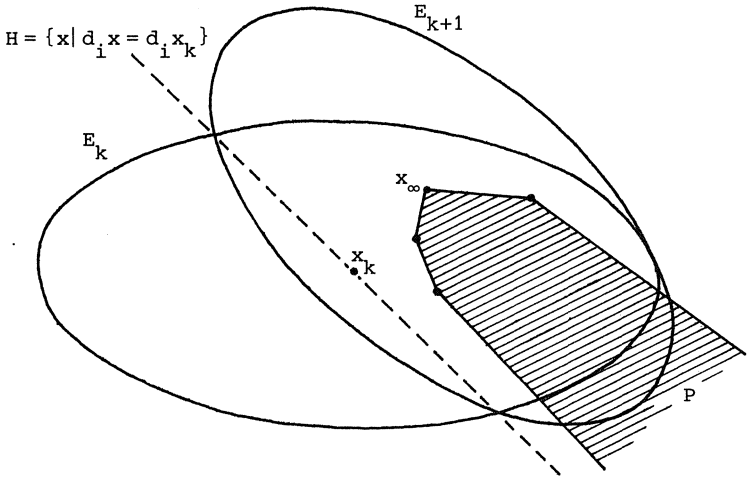
in het stelsel $Dx \leq b$ waaraan x_k niet voldoet, d.w.z.

(17) $\quad d_i x_k > b_i$.

Dan is E_{k+1} de ellipsoïde met kleinste volume zo dat

(18) $\quad E_{k+1} \supset E_k \cap \{x \mid d_i x \leq d_i x_k\}$

(vgl. Figuur 3).



Figuur 3.

Het blijkt dat een dergelijke kleinste ellipsoïde uniek is, en dat de bijbehorende A_{k+1} en x_{k+1} eenvoudig uit A_k, x_k en d_i berekend kunnen worden - zie §6. Uit (16), (17) en (18) volgt direct dat

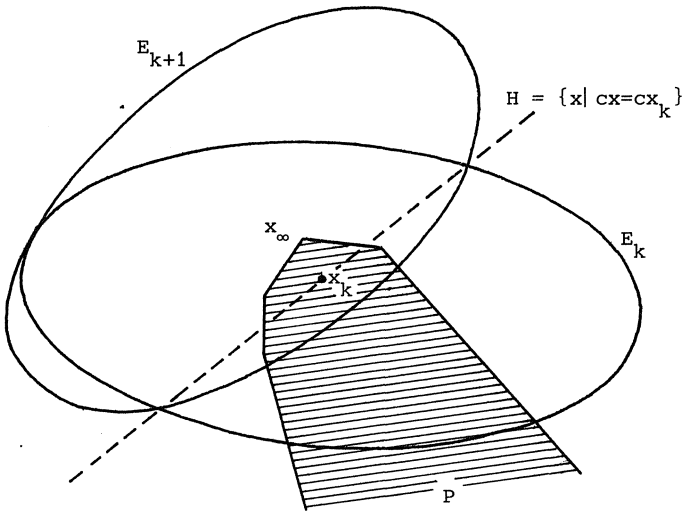
(19) $\quad P \cap E_k \subset P \cap E_{k+1}$.

Dus als $x_\infty \in E_k$ dan ook $x_\infty \in E_{k+1}$.

II. Als $x_k \in P$, dan wordt in principe dezelfde constructie uitgevoerd, maar nu met de doelvector c in plaats van d_i. Nu is E_{k+1} de kleinste ellipsoïde zo dat

$$E_{k+1} \supset E_k \cap \{x \mid cx \geq cx_k\} \tag{20}$$

(vgl. Figuur 4), welke ellipsoïde weer uniek is.



Figuur 4

Omdat $cx_\infty \geq cx_k$ en $x_\infty \in E_k$ volgt dat $x_\infty \in E_{k+1}$.

Zowel in geval I als in geval II hebben we de helft van E_k benaderd met een nieuwe ellipsoide E_{k+1}. Hoewel het volume van de nieuwe ellipsoïde groter is dan de helft van het volume van de oude ellipsoïde, kan worden aangetoond dat het volumen met een vaste factor < 1 afneemt:

$$\frac{\text{vol } E_{k+1}}{\text{vol } E_k} \leq \left(\frac{1}{2}\right)^{1/20n} \tag{21}$$

(zie §6). Omdat vol $E_0 \leq (2R)^n$, weten we dat

(22) $\quad \text{vol } E_N \leq \left(\frac{1}{2}\right)^{N/2On} . (2R)^n .$

Laat nu x_K het laatste element in de rij $x_0, \ldots, x_N$ zijn dat in P zit, d.w.z. zo dat geval II werd toegepast. Uit (19) volgt dat $x_K \in E_N$. Verder weten we dat $x_\infty \in E_N$. Het principe van de ellipsoïde-methode is nu dat vanwege (22) de ellipsoide E_N dermate klein is dat x_K een goede benadering voor x_∞ is. Bewezen kan worden dat

(23) $\quad \|x_K - x_\infty\| < \frac{1}{2Q^2}$

(zie §7). Omdat de noemers van de in x_∞ voorkomende coördinaten ten hoogste Q zijn, en omdat geen twee rationale getallen met noemers ten hoogste Q dichter dan $1/Q^2$ bij elkaar liggen, is x_∞ de enige vector, met noemers $\leq Q$, die aan (23) voldoet.

De coördinaten van x_∞ kunnen expliciet worden gevonden door de coördinaten van x_K stuk voor stuk te benaderen door een kettingbreukontwikkeling. De laatste benadering waarvan de noemer niet groter dan Q is, is de gezochte overeenkomstige coördinaat van x_∞.

Na deze ruwe schets van de ellipsoïde-methode moeten we het volgende nog preciseren:

- dat de vooronderstellingen zonder beperking van de algemeenheid mogen worden gemaakt - zie §5;
- hoe de ellipsoïde E_{k+1} uit de ellipsoïde E_k wordt bepaald - zie §6;
- dat de ellipsoïde E_N klein genoeg is om zeker te zijn van (23) - zie §7.

5. DE VOORONDERSTELLINGEN

We laten nu zien dat de in §3 gemaakte vooronderstellingen zonder beperking van de algemeenheid gemaakt mogen worden. De hieronder gegeven technieken om het oorspronkelijke LP-probleem te transformeren zijn van min of meer theoretisch belang. In de praktijk weet men vaak a priori dat aan de vooronderstellingen is voldaan, of kunnen deze op een eenvoudiger manier worden verkregen.

Ten eerste mogen we aannemen dat het polyeder P geen affiene deelruimten van dimensie ten minste 1, omvat. Als we hier niet zeker van zijn kunnen we (1) vervangen door

(24) $\quad \max \{cx-cx' \mid x,x' \geq 0,\ Dx - Dx' \leq b\}.$

Weliswaar verdubbelt de probleemgrootte hierdoor, maar deze blijft polynomiaal begrensd door de oorspronkelijke probleemgrootte.

Als P geen deelruimten omvat, dan wordt het maximum (1) bereikt in een hoekpunt van P (en misschien ook in andere punten van P), of is oneindig. Dus dan weten we dat, als M eindig is,

$$(25) \qquad |M| \leq nTQ \leq n^{2n}.T^{2n}.$$

Als we nu niet weten of P volledig-dimensionaal is, kunnen we (1) vervangen door

$$(26) \qquad \max \{cx - (3n^{3n}T^{3n})\lambda \mid \lambda \geq 0,\ -\lambda + Dx \leq b\},$$

waarbij λ een nieuwe variable is. Dan omvat het nieuwe polytoop

$$(27) \qquad P' = \{(x,\lambda) \mid \lambda \geq 0,\ -\lambda + Dx \leq b\}$$

weer geen deelruimten, dus (26) wordt bereikt in een hoekpunt (x_0,λ_0) van P'. Weer kan eenvoudig worden ingezien dat de tellers en noemers van de hoekpunten van P' ten hoogste Q in absolute waarde zijn. Dus als $\lambda_0 > 0$, dan is $\lambda_0 \geq 1/Q$, en is (26) ten hoogste

$$(28) \qquad nTQ - \frac{3n^{3n}T^{3n}}{Q} \leq -2n^{2n}T^{2n}.$$

Dus als (26) groter is dan $-2n^{2n}T^{2n}$, dan wordt dit maximum bereikt in een hoekpunt (x_0,λ_0) met $\lambda_0 = 0$, en is x_0 een oplossing voor (1). Als (26) kleiner is dan $-2n^{2n}T^{2n}$, dan is, volgens (25), $\lambda > 0$, en dan bestaat er kennelijk geen x met $Dx \leq b$. Bovendien kan eenvoudig worden ingezien dat (26) eindig is als en alleen als (1) eindig is.

Merk op dat het polyeder P' volledig-dimensionaal is, en dat de afmeting van het LP-probleem (26) polynomiaal begrensd wordt door de afmeting van het oorspronkelijke LP-probleem.

We mogen dus aannemen dat P volledig-dimensionaal is en geen affiene deelruimten omvat. Ook mogen we aannemen dat P begrensd is. Als we dit niet weten, kunnen we probleem (1) vervangen door de problemen: bepaal

$$(29) \qquad M_1 = \max \{cx \mid Dx \leq b,\ -2Q \leq x^i \leq +2Q \ (i = 1,\ldots,n)\} \text{ en}$$
$$M_2 = \max \{cx \mid Dx \leq b,\ -3Q \leq x^i \leq +3Q \ (i = 1,\ldots,n)\}.$$

Eenvoudig kan worden ingezien dat als $M_1 = M_2$ dan is $M = M_1$, en als $M_2 > M_1$ dan is $M = \infty$.

Tenslotte mogen we aannemen dat het maximum (1) in precies één hoekpunt van P wordt aangenomen. Vervang anders c door

$$(30) \qquad c' = S^n.c + (1,S,S^2,\ldots,S^{n-1}),$$

waarbij $S = 4.Q^{3n}$. Dan wordt $\max\{c'x \mid x \in P\}$ in precies één hoekpunt van P aangenomen, zeg in x_∞, en bovendien geldt dat $M = cx_\infty$.

Aangezien elk van deze transformaties in polynomiaal-begrensde tijd kan worden uitgevoerd, kan het oorspronkelijke LP-probleem in polynomiale tijd worden omgevormd tot een probleem dat aan de genoemde vooronderstellingen voldoet.

6. DE KLEINSTE ELLIPSOIDE

We laten nu zien hoe de parameters A_{k+1} en x_{k+1} voor de ellipsoïde E_{k+1} berekend kunnen worden uit die voor E_k, en dat het volume van E_{k+1} klein genoeg is ten opzichte van dat van E_k.

De ellipsoïde E_k werd gegeven door

$$(31) \qquad E_k = \{x \mid (x-x_k)^T A_k^{-1} (x-x_k) \leq 1\}.$$

Verder werd een halfruimte H gegeven door, zeg,

$$(32) \qquad H = \{x \mid ax \geq ax_k\},$$

voor zekere vector a ($a = -d_i$ en $a = c$ in geval I, resp. II). Nodig is de kleinste ellipsoïde E_{k+1} te bepalen zo dat

$$(33) \qquad E_{k+1} \supset E_k \cap H.$$

Nu bestaat er, zoals eerder werd opgemerkt, een affiene transformatie die de ellipsoïde E_k overbrengt naar de eenheidsbol $\{x \mid x^T x \leq 1\}$, en de halfruimte H naar de halfruimte $\{x \mid x^1 \geq 0\}$ (waarbij $x = (x^1,x^2,\ldots,x^n)$). Het is een eenvoudige meetkundige opgave de unieke kleinste ellipsoïde E te bepalen zo dat

$$(34) \qquad E \supset \{x \mid x^T x \leq 1,\ x^1 \geq 0\}.$$

Omdat de volumens van meetkundige figuren onder een affiene transformatie met een constante factor worden vermenigvuldigd (nl. met de absolute waarde van de determinant van de bijbehorende matrix), krijgen we door de inverse affiene transformatie op E toe te passen de kleinste ellipsoïde E_{k+1} die $E_k \cap H$ omvat.

Uitwerking geeft de volgende formules voor A_{k+1} en x_{k+1}:

$$A_{k+1} = \frac{n^2}{n^2-1}\left(A_k - \frac{2}{n+1} b_k b_k^T\right), \tag{35}$$

$$x_{k+1} = x_k + \frac{1}{n+1} b_k, \tag{36}$$

waarbij

$$b_k = A_k a / \sqrt{a^T A_k a}. \tag{37}$$

Nu weten we dat

$$\frac{\text{vol } E_{k+1}}{\text{vol } E_k} = \sqrt{\frac{\det A_{k+1}}{\det A_k}} = \left(\frac{n^2}{n^2-1}\right)^{\frac{1}{2}n} \left(\frac{n-1}{n+1}\right)^{\frac{1}{2}} \leq (\tfrac{1}{2})^{1/20n} \tag{38}$$

omdat we zonder beperking der algemeenheid mogen aannemen dat $A_k = I$ en $a = (1,0,\ldots,0)^T$. Uit (38) volgt

$$\text{vol } E_N \leq (\tfrac{1}{2})^{N/20n} \cdot (2R)^n \tag{39}$$

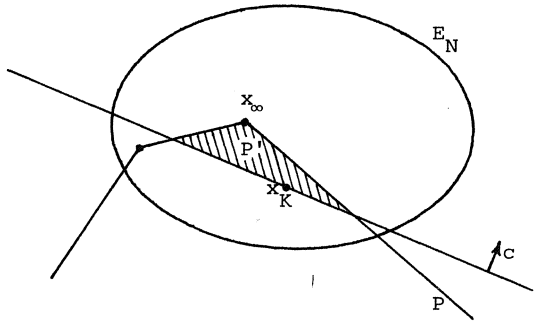
Er schuilt een adder onder het gras: om b_k te berekenen moeten we de wortel trekken. Dit impliceert dat we bij toepassing van de methode op een eindige precisie-computer moeten afronden, en dat we daardoor de ellipsoïde E_{k+1} iets ruimer moeten nemen om afrondingsfouten te neutraliseren. Het zal blijken dat we kunnen volstaan met een afronding tot op een polynomiaal aantal bits, en dat de iets ruimer gekozen ellipsoïden ook de gewenste polynomiale algorithme voor lineaire programmering opleveren - zie §8.

7. E_N IS KLEIN GENOEG

We laten nu zien dat E_N inderdaad klein genoeg is om te weten dat de afstand tussen de vectoren x_K en x_∞ kleiner dan $1/2Q^2$ is.

We weten dat $x_K \in E_N$, $x_\infty \in E_N$ en dat

$$P' := P \cap \{x \mid cx \geq cx_K\} \subset E_N. \tag{40}$$



Figuur 5

We bewijzen dat alle punten van P' dichter dan $1/2Q^2$ bij x_∞ liggen. Want stel dat een punt $x'_K \neq x_\infty$ in P' bestaat zo dat

(41) $\quad \|x'_K - x_\infty\| \geq 1/2Q^2$.

Zonder beperking der algemeenheid mogen we aannemen dat x'_K een hoekpunt van P' is. Als P' nog andere hoekpunten van P bevat behalve x_∞, dan kunnen we voor x'_K een hoekpunt van P ongelijk x_∞ nemen zo dat cx'_K zo groot mogelijk is (hoekpunten van P liggen nooit dichter dan $1/2Q^2$ bij elkaar).

Laat $D'x \leq b'$ het stelsel van die ongelijkheden uit $Dx \leq b$ zijn die voor $x = x_\infty$ overgaan in gelijkheid. Bekijk het polytoop

(42) $\quad P'' = \{x \mid D'x \leq b',\ cx \geq cx_\infty - 1\}$.

P'' is begrensd omdat $\max\{cx \mid x \in P\}$ in precies één hoekpunt van P wordt aangenomen. P'' is volledig-dimensionaal omdat P volledig-dimensionaal is.

Uit een redenering analoog aan die gemaakt in §3 volgt dat de tellers en de noemers van de coördinaten van de hoekpunten van P'' in absolute waarde niet groter dan $n^{2n^2}T^{2n^2}$ zijn. Hieruit volgt ten eerste dat

(43) $\quad \text{vol } P'' \geq (n^{5n^4}.T^{4n^4})^{-1}$,

omdat P'' n+1 affien onafhankelijke hoekpunten heeft, waarvan het convex omhulsel een volumen heeft gelijk aan 1/n! maal de determinant van een matrix

met noemers ten hoogste $n^{4n^2}T^{4n^2}$, dus dit volumen is ten minste het rechterlid van (43). Ten tweede volgt dat ieder tweetal hoekpunten van P", en dus ook ieder tweetal willekeurige punten in P", een afstand ten hoogste

(44) $\qquad 2n^{3n^2}.T^{2n^2}$

hebben. Omdat

(45) $\qquad P' \supset \{x \mid D'x \leq b',\ cx \geq cx_\infty - (cx_\infty - cx'_K)\},$

volgt dat

(46) $\qquad \text{vol } P' = (cx_\infty - cx'_K)^n.\text{vol } P'' \geq (cx_\infty - cx'_K)^n (n^{5n^4}.T^{4n^4})^{-1}.$

Evenzo bevat het rechterlid van (45) geen twee punten op afstand groter dan

(47) $\qquad (cx_\infty - cx'_K)2n^{3n^2}T^{2n^2}.$

In het bijzonder volgt de tegenspraak

(48) $\qquad \frac{1}{2Q^2} \leq \|x_\infty - x'_K\| \leq (cx_\infty - cx'_K)2n^{3n^2}T^{2n^2} \leq 2n^{3n^2}T^{2n^2}n^{5n^3}T^{4n^3}(\text{vol } P')^{1/n} \leq$

$\leq 2n^{8n^3}T^{6n^3}(\text{vol } E_N)^{1/n} \leq 2n^{8n^3}T^{6n^3}(1/2)^{N/20n^2}\ 2R \leq 4n^{11n^3}T^{8n^3}(1/2)^{N/20}$

$< \frac{1}{2Q^2},$

gebruik makend van resp. (41), (47), (46), (40), (39), (11) en (10).

8. PRECISIE EN PRAKTISCHE TOEPASBAARHEID

In §6 merkten we op dat een expliciete, exacte uitvoering van de ellipsoïde-methode zoals hierboven beschreven op een eindige precisie-computer niet mogelijk is omdat bij de overgang van E_k op E_{k+1} wortel moet worden getrokken. Dit kan worden opgelost door alle berekeningen uit te voeren tot op

(49) $\qquad p := 5N\lceil \log \frac{12\sqrt{n}}{R^2} \rceil$

binaire bits nauwkeurig, door de ellipsoïden iets ruimer te kiezen, nl. door A_{k+1} in plaats van als in (35) als volgt te geven

$$(50) \qquad A_{k+1} = \frac{2n^2+3}{2n^2}\left(A_k - \frac{2}{n+1} b_k b_k^T\right),$$

en door N te verhogen tot

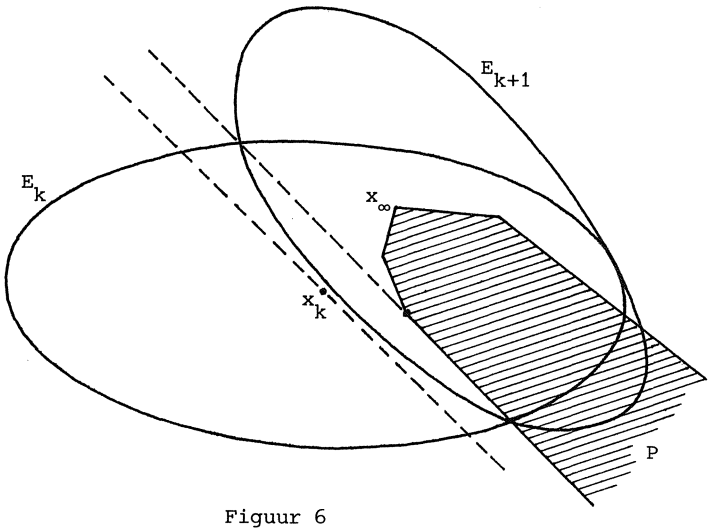
$$(51) \qquad 20\lceil n^5.\log 12 + 13n^8.\log n + 10n^8.\log T\rceil$$

Dit wordt bewezen in [18]. Merk op dat p en N begrensd zijn door een polynoom in de afmeting van het LP-probleem, zodat de hele methode uitvoerbaar is binnen polynomiaal begrensde tijd.

Toch zijn zowel het aantal stappen N als de vereiste nauwkeurigheid p veel te groot voor de computers van dit moment om de ellipsoïde-methode in deze vorm ook maar enige praktische toepasbaarheid toe te schrijven. Weliswaar zijn hierboven, ter bevordering van de eenvoud, de berekeningen aan de ruime kant uitgevoerd, maar ook scherpere afschattingen van N en p lijken de methode niet essentieel te verbeteren.

Er zijn verschillende wijzigingen voorgesteld om de efficiëntie en de numerieke stabiliteit van de methode te verhogen, zoals:

(i) In plaats van de ellipsoïde E_k te "halveren" d.m.v. een hypervlak door het middelpunt x_k, en voor E_{k+1} de kleinste ellipsoïde te nemen die de "goede" helft van E_k omvat, kunnen we het hypervlak opschuiven tot tegen P.



Figuur 6

D.w.z., als geval I zich voordoet (cf. §4), en $d_i x \leq b_i$ is een vergelijking uit het stelsel $Dx \leq b$ zo dat $d_i x_k > b_i$, dan is E_{k+1} de kleinste ellipsoïde met

$$(52) \qquad E_{k+1} \supset E_k \cap \{x \mid d_i x \leq b_i\}.$$

De bij deze "diepe snede-methode" behorende formules voor A_{k+1} en x_{k+1} zijn bepaald door Padberg en Rao [33] en verschillende anderen (zie Wolfe [38]). Men kan deze methode nog verscherpen door niet één ongelijkheid uit het stelsel $Dx \leq b$ te kiezen, maar een niet-negatieve lineaire combinatie $d'x \leq b'$ van ongelijkheden uit $Dx \leq b$, zo dat $d'x_k > b'$, en zo dat $E_{k+1} \supset$ $\supset E_k \cap \{x \mid d'x \leq b'\}$ zo klein mogelijk gekozen kan worden. Dit is de methode van de "surrogaat sneden" - zie Goldfarb en Todd [16]. Bland, Goldfarb en Todd [2] lieten zien dat ook deze wijzigingen niet leiden tot een essentiële verbetering van de ellipsoïde-methode.

(ii) Als te grof wordt afgerond, d.w.z. als p te klein wordt gekozen, dan hoeven de matrixen A_k niet positief-definiet te blijven, zodat de deling door $\sqrt{a^T A_k a}$ als in (37) niet steeds uitvoerbaar is. Dit kan verholpen worden door A_k te schrijven als $A_k = J_k J_k^T$, waarbij J_k een niet-singuliere matrix is, en door de rij $A_0, A_1, \ldots$ te vervangen door $J_0, J_1, \ldots$. Zie Bland, Goldfarb en Todd [2] voor de expliciete formules.

Daarnaast zijn allerlei verbeteringen mogelijk door het onderhavige LP-probleem nader te beschouwen. Vaak zal bijvoorbeeld de begin-ellipsoïde E_0 zuiniger gekozen kunnen worden dan hierboven, doordat het polytoop P a priori beter begrensd kan worden.

Maar basisidee van de ellipsoïde-methode is te eindigen met een zeer kleine ellipsoïde E_N, hetgeen als basisproblemen een grote precisie en een groot aantal halveringen onvermijdelijk lijken te maken.

Een probleem is ook dat men in het algemeen niet eerder dan na N stappen zeker is dat men dichtbij het optimum x_∞ is. Een eenvoudige omvorming van de methode geeft ons echter ook een polynomiale methode om te beslissen of het open polyeder

$$(53) \qquad P^o = \{x \mid Dx < b\}$$

leeg is of niet, en zo niet, een punt in P^o te vinden. Eveneens kan ieder LP-probleem worden teruggebracht tot het probleem een oplossing voor een

stelsel strikte lineaire ongelijkheden $Dx < b$ te vinden. Het voordeel van deze alternatieve formulering is dat vaak niet alle N stappen hoeven worden uitgevoerd, maar dat gestopt kan worden zodra een x_k gevonden is met $Dx_k < b$. In feite is dit de oorspronkelijke methode van Khachian [23] - zie Gács en Lovász [11]. Vergelijkt men de ellipsoïde-methode met de simplex-methode dan lijkt het daarom niet helemaal eerlijk de zeer grote N te vergelijken met de $\frac{3}{2}m$ pivot-stappen van de simplex-methode, d.w.z. een theoretische bovengrens voor het "worst case"-gedrag van de ellipsoïde-methode met het praktische "average case"-gedrag van de simplex-methode.

Het moet overigens niet uitgesloten worden geacht dat een combinatie van simplex- en ellipsoïde-methode nog eens zal leiden tot een zowel in praktisch als in theoretisch opzicht efficiënte algorithme voor lineaire programmering. De simplex-methode heeft het voordeel dat de bezochte punten "mooi" blijven, nl. hoekpunten van het polyeder, maar het nadeel dat een verkeerde keuze van de reisrichting (d.w.z. van de pivot-elementen) tot lange reizen kan leiden. Omgekeerd heeft de ellipsoïde-methode het voordeel dat de gevolgde reisrichting in polynomiale tijd tot het optimum leidt, maar het nadeel dat de punten $x_0, x_1, \ldots$ grote noemers kunnen hebben. De vraag is dus: kunnen de beide voordelen gecombineerd worden?

9. OPTIMALISERINGS- EN SCHEIDINGSALGORITHMEN

In de inleiding werd al opgemerkt dat het toepassingsgebied van de ellipsoïde-methode ruimer is dan lineaire programmering. In de rest van dit artikel geven we hiervan een ruwe schets.

Stel dat een begrensde, gesloten, volledig-dimensionale, convexe verzameling P in $\mathbb{Q}^n$ is gegeven, en dat we het volgende probleem willen oplossen.

Optimaliseringsprobleem. Gegeven $c \in \mathbb{Q}^n$, bepaal $\max\{cx \mid x \in P\}$.

Nadere beschouwing van de hierboven gegeven ellipsoïde-methode leert dat dit optimaliseringsprobleem kan worden benaderd in polynomiaal begrensde tijd als het volgende probleem kan worden opgelost in polynomiaal begrensde tijd.

Scheidingsprobleem. Gegeven $y \in \mathbb{Q}^n$, bepaal of $y \in P$, en zo niet, vind een $a \in \mathbb{Q}^n$ zo dat $ay > \max\{ax \mid x \in P\}$.

Dit scheidingsprobleem vraagt dus om een hypervlak dat y van P scheidt. We kunnen een polynomiale algorithme voor het scheidingsprobleem invoeren als subroutine in de ellipsoïde-methode, nl. om te beslissen tussen de gevallen I en II uit §4, en om in geval I een halveringshypervlak te vinden. Omdat de ellipsoïde-methode niet meer vraagt over P dan dit, geeft dit ons een polynomiale algorithme om het optimaliseringsprobleem te benaderen. (In feite zou een convexe verzameling P in de computer gedeclareerd kunnen worden als een algorithme voor het scheidingsprobleem.) Dit wordt precieser geformuleerd en bewezen in [18]. Noodzakelijk is dan een punt a_0 en getallen r en R te weten zo dat

(54) $\quad S(a_0,r) \subset P \subset S(a_0,R),$

waarbij $S(a_0,\rho)$ de bol om a_0 met straal ρ voorstelt. "Polynomiaal" betekent: de looptijd wordt begrensd door een polynoom in $|\log r|$, $|\log R|$, $\log T$ en $|\log \varepsilon|$, waarbij T het grootste getal is dat voorkomt in tellers en noemers van a_0, c of y (in absolute waarde), en waarbij ε de nauwkeurigheid is die het antwoord moet hebben. D.w.z., het optimaliseringsprobleem vraagt om een x_∞ zo dat $d(x_\infty,P) < \varepsilon$ en $cx_\infty + \varepsilon > \max\{cx \mid x \in P\}$, en het scheidingsprobleem vraagt te bepalen of $d(y,P) < \varepsilon$, of een a in $\mathbb{Q}^n$ te vinden zo dat $ay + \varepsilon > \max\{ax \mid x \in P\}$.

De ellipsoïde-methode geeft ons dus de implicatie:

(55) $\quad \exists$ polynomiale scheidingsalgorithme $\Rightarrow \exists$ polynomiale optimaliseringsalgorithme.

De implicatie geldt echter ook in de omgekeerde richting. We kunnen zonder beperking der algemeenheid aannemen dat het hierboven omschreven punt a_0 gelijk is aan de oorsprong $\underline{0}$. Als we nu P vervangen door de duale convexe verzameling

(56) $\quad P^* = \{y \mid xy \leq 1 \text{ voor iedere } x \text{ in } P\},$

dan blijken scheidings- en optimaliseringsprobleem in elkaar over te gaan. Dus dan:

(57) $\quad \exists$ polynomiale optimaliseringsalgorithme voor P $\Rightarrow \exists$ polynomiale scheidingsalgorithme voor $P^* \Rightarrow \exists$ polynomiale optimaliseringsalgorithme voor $P^* \Rightarrow \exists$ polynomiale scheidingsalgorithme voor P,

en daarom:

(58) $\exists$ polynomiale scheidingsalgorithme voor P $\Leftrightarrow \exists$ polynomiale optimaliseringsalgorithme voor P.

In de volgende paragrafen geven we een paar toepassingen van dit principe.

10. KWADRATISCHE PROGRAMMERING

Een van de verschijningsvormen van een kwadratisch programmeringsprobleem (QP-probleem) is: bepaal

(59) $\min \{x^T Bx + cx \mid Dx \leq b\}$.

Hierin is $Dx \leq b$ weer een stelsel lineaire ongelijkheden, en cx het inwendig product van c en x, en B is een symmetrische positief semi-definiete $n \times n$-matrix. Weer mogen we zonder de algemeenheid te beperken aannemen dat B, c, D en b geheeltallig zijn. We kunnen (59) benaderen met de ellipsoïde-methode als volgt. Het QP-probleem is equivalent met: bepaal

(60) $\min \{\lambda \mid Dx \leq b,\ \lambda \geq x^T Bx + cx\}$.

Als we definieren

(61) $P = \{(x,\lambda) \mid Dx \leq b,\ \lambda \geq x^T Bx + cx\}$,

dan is P een gesloten convexe verzameling, en is (60) een speciaal geval van het optimaliseringsprobleem voor P. De verzameling P kan worden opgevat als de grafiek van de functie $x^T Bx+cx$ gedefinieerd op $\{x \mid Dx \leq b\}$, aangevuld met alle punten die boven de grafiek liggen. We kunnen P begrensd maken door P te vervangen door $P \cap \{(x,\lambda) \mid \lambda \leq U\}$, waarbij U groot genoeg wordt gekozen (in het algemeen zal het niet moeilijk zijn een dergelijke U te vinden).

Om (60) in polynomiale tijd te vinden is het volgens de voorgaande paragraaf voldoende te laten zien dat het scheidingsprobleem voor P in polynomiale tijd kan worden opgelost. Dit laatste is niet moeilijk. Als

we een (x',λ') kiezen dan kan door substitutie eenvoudig worden nagegaan of $(x',\lambda') \in P$. Als blijkt dat (x',λ') niet in P zit, dan is of aan $Dx' \leq b$ niet voldaan, of aan $\lambda' \geq x'^{T}Bx'+cx'$. In het eerste geval levert de ongelijkheid waaraan niet is voldaan een scheidend hypervlak. In het tweede geval, d.w.z. als $\lambda' < x'^{T}Bx'+cx'$, scheidt het hypervlak

$$(62) \qquad \{(x,\lambda) \mid (x'^{T}B+c)x - \lambda = (x'^{T}Bx'+cx'-\lambda')\}$$

het punt (x',λ') van P, en kunnen we als oplossing voor het scheidingsprobleem de vector $a = (x'^{T}B+c,-1)$ nemen.

Het scheidingsprobleem voor P kan dus in polynomiale tijd worden opgelost, en dus ook het optimaliseringsprobleem, en daarom bestaat een polynomiale algorithme voor (convexe) kwadratische programmering. Voor een preciesere beschrijving verwijzen we naar Kozlov, Tarasov en Khachian [26].

11. TOEPASSINGEN IN DE COMBINATORISCHE OPTIMALISERING

In [18], [22] en [35] wordt aangetoond hoe de ellipsoïde-methode ook in de combinatorische optimalisering tot een aantal nieuwe inzichten leidt. Een voorbeeld hiervan is het volgende.

Stel we hebben een gerichte graaf $G = (V,E)$, waarin een vast punt r is gekozen. Een *r-vertakking* is een verzameling E' van kanten van G zo dat ieder punt van G bereikbaar is vanuit r via een gericht pad van kanten in E'. (Dus de minimale r-vertakkingen zijn precies de in r gewortelde gerichte bomen.) Nu is het een interessante vraag om, gegeven een "lengte" functie $\ell: E \longrightarrow \mathbb{Z}_+$, een r-vertakking E' te vinden met minimale totale lengte, d.w.z. met

$$(63) \qquad \sum_{e \in E'} \ell(e)$$

zo klein mogelijk. Er bestaat een polynomiale algorithme van Fulkerson [10] voor dit probleem, maar met de ellipsoïde-methode kan de polynomiale oplosbaarheid van dit probleem worden herleid tot die van een eenvoudiger probleem.

Definieer

(64) P := het convex omhulsel van de karakteristieke functies van de r-vertakkingen.

P is dus een convex polytoop in $\mathbb{Q}^E$, en het bovenbeschreven minimaliseringsprobleem is equivalent met

(65) $\quad \min\{\ell x \mid x \in P\}.$

Volgens het principe beschreven in §9 kan dit minimum in polynomiaal-begrensde tijd worden bepaald, als er een polynomiale scheidingsalgorithme voor P bestaat. Nu heeft Fulkerson ook de volgende karakterisering van P gegeven:

$$(66) \qquad P = \{x \in \mathbb{Q}^E \mid 0 \leq x(e) \leq 1 \quad (e \in E),\ \sum_{e \in E'} x(e) \geq 1 \quad (E' \text{ r-snede})\},$$

waarbij een verzameling E' van kanten van G een r-*snede* heet als er een niet-lege verzameling V' van punten van G bestaat zo dat $r \notin V'$ en E' is de verzameling kanten van $V \backslash V'$ naar V'.

Het is duidelijk dat P bevat moet zijn in het rechterlid van (66), omdat de karakteristieke vector van iedere r-vertakking voldoet aan de lineaire ongelijkheden. Fulkerson's stelling geeft de omgekeerde inclusie.

Dus om het scheidingsprobleem voor P op te lossen moeten we, gegeven een x in $\mathbb{Q}^E$, nagaan of x aan de ongelijkheden in (66) voldoet. Eenvoudig is in polynomiale tijd na te gaan of $0 \leq x(e) \leq 1$ voor iedere kant e. Als er een kant e bestaat die hieraan niet voldoet dan geeft deze kant ons een scheidend hypervlak. Om het tweede stelsel ongelijkheden na te gaan kunnen we niet alle r-sneden E' aflopen en $\sum_{e \in E'} x(e) \geq 1$ controleren, omdat er exponentieel veel (ongeveer $2^{|V|-1}$) r-sneden bestaan. Toch kan dit stelsel in polynomiale tijd worden nagegaan, en wel als volgt. Vat de functie x op als een capaciteitsfunctie op de kanten van G. We kunnen nu zoeken naar een r-snede met minimale capaciteit. Als deze minimale capaciteit ten minste 1 bedraagt besluiten we tot: $x \in P$. Zo niet, dan levert de r-snede met capaciteit kleiner dan 1 ons een scheidend hypervlak.

- Een r-snede met minimale capaciteit kan worden gevonden door voor ieder punt $s \neq r$ een snede E_s van minimale capaciteit te bepalen die r van s scheidt. Deze kan worden gevonden met de minimum-snede algorithme van Ford en Fulkerson [9]. Kiezen we die snede uit $\{E_s \mid s \neq r\}$ met de laagste capaciteit dan hebben we een r-snede met minimale capaciteit.

Zo wordt de polynomiale oplosbaarheid van het "minimum lengte van een r-vertakking"-probleem afgeleid uit die van het "minimum capaciteit van een r-snede"-probleem. Nu kan op analoge wijze omgekeerd worden laten zien dat ook de polynomiale oplosbaarheid van het laatste probleem afgeleid kan wor-

den uit die van het eerste. Op deze manier vinden we een zekere dualiteit tussen combinatorische problemen. Zo blijkt de polynomiale oplosbaarheid van ieder van de onderstaande linkerproblemen equivalent met de polynomiale oplosbaarheid van de rechterproblemen.

(i) min.lengte van een r-vertakking; (ii) min.capaciteit van een r-snede;
(iii) min.lengte van een pad van r naar s; (iv) min.capaciteit van een r-s-snede;
(v) min.gewicht van een perfecte matching; (vi) min.capaciteit van een oneven snede.

(De laatste twee problemen hebben betrekking op een ongerichte graaf G = (V,E) met V even. Een *perfecte matching* is een verzameling disjuncte kanten van G die V overdekken. Een *oneven snede* is de collectie kanten van V' naar V\V', waarbij V' een oneven verzameling punten van G is.)

Voor probleem (iii) is een eenvoudige polynomiale algorithme bekend ontworpen door Dijkstra [7], terwijl probleem (iv) kan worden opgelost met de iets moeilijkere algorithme van Ford en Fulkerson [9]. Voor probleem (v) is een vrij gecompliceerde algorithme opgesteld door Edmonds [8], terwijl probleem (vi) kan worden opgelost door een eenvoudige aanpassing van de "minimum snede"-algorithme van Ford en Fulkerson (zie Padberg en Rao [34]).

Voor meer toepassingen van dit dualiteitsprincipe verwijzen we naar [18]. Weliswaar kunnen de meeste van deze toepassingen, net als het bovenstaande r-vertakkingsprobleem, worden geformuleerd als een LP-probleem (en lijken dus op het eerste gezicht direct oplosbaar met Khachian's methode), maar de formulering van deze LP-problemen vergt in het algemeen al exponentieel veel tijd vanwege het grote aantal lineaire ongelijkheden (bijvoorbeeld r-sneden).

De toepassingen berusten sterk op de polyhedrale karakteriseringen van zekere polytopen, zoals de karakterisering (66) van het convex omhulsel P van de r-vertakkingen. Opgemerkt moet worden dat dergelijke polyhedrale karakteriseringen vaak verkregen werden als bijproduct van de constructie van een polynomiale algorithme. Nu met de ellipsoïde-methode is het juist omgekeerd: een polyhedrale karakterisering kan aanleiding geven tot een polynomiale algorithme, hetgeen verder onderzoek naar dergelijke karakteriseringen motiveert.

Overigens zijn de aldus met de ellipsoïde-methode gevonden algorithmen weliswaar polynomiaal, maar verre van efficiënt, vanwege de hoge graad van het polynoom dat de looptijd begrenst. Zij vormen dan ook geen alter-

natief voor de eerder ontwikkelde speciale algorithmen. In de volgende paragraaf zullen we echter laten zien dat met de ellipsoïde-methode ook de polynomiale oplosbaarheid van een aantal combinatorische problemen kan worden afgeleid die nog niet eerder in deze zin waren opgelost.

12. PERFECTE GRAFEN EN SUBMODULAIRE FUNCTIES

We zullen nu ingaan op twee combinatorische problemen waarvan de polynomiale oplosbaarheid alleen werd aangetoond met de ellipsoïde-methode. Uitwerking levert weer algorithmen die, hoewel polynomiaal, een zeer lage graad van efficiëntie hebben. Het bewijs van de polynomiale oplosbaarheid kan veeleer worden beschouwd als een motivering om, nu we eenmaal weten dat polynomiale algorithmen bestaan, te zoeken naar werkelijk efficiënte algorithmen. Hierbij is het onwaarschijnlijk dat een verscherping van de ellipsoïde-methode tot essentiële verbeteringen zal leiden. Voor de details verwijzen we naar [18].

Perfecte grafen. Een ongerichte graaf $G = (V,E)$ heet *perfect* als voor iedere geïnduceerde deelgraaf G' van G geldt:

$$\omega(G') = \chi(G'). \tag{67}$$

Hierbij is $\omega(G')$ het *kliekgetal* van G' (d.w.z. de afmeting van de grootste kliek in G'), en $\chi(G')$ het *kleurgetal* van G' (d.w.z. het minimale aantal kleuren dat nodig is om de punten van G' zo te kleuren dat ieder tweetal verbonden punten verschillend gekleurd zijn). Eenvoudig is in te zien dat $\omega(G) \leq \chi(G)$ voor iedere graaf G.

Het probleem om van een willekeurige graaf het kliekgetal (of het kleurgetal) te bepalen is NP-volledig (zie Garey en Johnson [13] en Van Leeuwen [28]). Voor verschillende deelklassen van de klasse der perfecte grafen bestaan echter polynomiale algorithmen om het kliekgetal te bepalen (zoals voor bipartite grafen, lijngrafen van bipartite grafen, getrianguleerde grafen, transitief-orientieerbare grafen, en hun complementen). Het was een open vraag of voor de klasse van *alle* perfecte grafen zo'n algorithme bestaat. Deze vraag kan bevestigend beantwoord worden met behulp van de ellipsoïde-methode. Omdat, zoals Lovász [29] bewees, het complement van een perfecte graaf weer perfect is, geeft dit tegelijk een polynomiale algorithme om het onafhankelijkheidsgetal $\alpha(G)$ van een perfecte graaf G te vinden ($\alpha(G)$ = het maximale aantal paarsgewijs niet-ver-

bonden punten in G).

Lovász [30] liet zien dat als G = (V,E) een perfecte graaf is, dan is

(68) $$\omega(G) = \max\{ \sum_{i,j=1}^{n} a_{ij} \mid A = (a_{ij})_{i,j=1}^{n} \text{ is een positief semi-definiete matrix zo dat Tr } A = 1 \text{ en } a_{ij} = 0 \text{ als } i \text{ en } j \text{ verschillende niet-verbonden punten zijn}\},$$

waarbij is aangenomen dat V = {1,...,n}. Het bepalen van dit maximum is een speciaal geval van het optimaliseringsprobleem voor de convexe verzameling P van alle matrixen A die voldoen aan de voorwaarden vermeld in (68). Volgens het principe uit §9 is dit probleem in polynomiale tijd oplosbaar als een polynomiale scheidingsalgorithme voor P bestaat. Nu kan, gegeven een matrix $A = (a_{ij})_{i,j=1}^{n}$, eenvoudig worden nagegaan of Tr A = 1 en of $a_{ij} = 0$ als i en j verschillend en niet verbonden zijn. Als aan een van deze voorwaarden niet voldaan is kan op eenvoudige wijze een scheidend hypervlak worden bepaald. Ook of A positief semi-definiet is kan worden gecontroleerd binnen polynomiale tijd, en wel als volgt. Vind een hoofddeelmatrix A' van A (d.w.z. een deelmatrix symmetrisch om de hoofddiagonaal van A) met rang A' = rang A. Een dergelijke matrix kan worden gevonden met de gebruikelijke Gauss-eliminatie. Dan is A positief semi-definiet als en alleen als A' positief definiet is. Zonder beperking der algemeenheid is $A' = (a_{ij})_{i,j=1}^{k}$. Nu is A' positief definiet als en alleen als

(69) $$\det((a_{ij})_{i,j=1}^{k'}) > 0$$

voor k' = 1,...,k. Deze determinanten kunnen in polynomiale tijd worden uitgerekend. Bovendien, als een dezer determinanten niet positief is, dan kan hieruit een hypervlak worden gevonden dat A van P scheidt.

Het scheidings- en het optimaliseringsprobleem voor P zijn dus polynomiaal oplosbaar, en dus kan $\omega(G)$ in polynomiale tijd worden bepaald. In [18] wordt aangetoond hoe hieruit ook polynomiale algorithmen kunnen worden afgeleid om expliciet een kliek ter grootte $\omega(G)$, en een goede kleuring met $\chi(G)$ kleuren te vinden.

Submodulaire functies. Een tweede toepassing van de ellipsoïde-methode is het vinden van de minimale waarde van een zgn. submodulaire functie. Een functie f gedefinieerd op de deelverzamelingen van een eindige verza-

meling X heet *submodulair* als

(70) $\quad f(X') + f(X'') \geq f(X' \cap X'') + f(X' \cup X'')$

voor ieder tweetal deelverzamelingen X',X'' van X. Voorbeelden van submodulaire functies zijn: (i) X is de verzameling rijen van een matrix, en f(X') is de rang van de rijen in X'; (ii) X is de verzameling punten van een gerichte graaf G = (X,E), $c:E \longrightarrow \mathbb{Z}_+$ is een "capaciteits"-functie, en f(X') is de totale capaciteit van de kanten die X' verlaten. Weliswaar wordt de minimum waarde van deze submodulaire functies trivialerwijs door $X' = \emptyset$ aangenomen, maar interessantere problemen kunnen worden afgeleid uit deze functies. Bijvoorbeeld, gegeven een gewichtsfunctie $w:X \longrightarrow \mathbb{Q}$, kan het minimum van $f(X') - \sum_{x \in X'} w(x)$ gevonden worden (dit definieert weer een submodulaire functie). Verder, als de submodulaire functie g gedefinieerd wordt door

(71) $\quad g(X') = f(X' \cup \{r\})$

voor $X' \subset X \setminus \{r,s\}$, dan is, voor voorbeeld (ii), het vinden van de minimale waarde van g equivalent met het vinden van een r-s-snede van minimale capaciteit.

We moeten voorzichtig zijn met de manier waarop de functie f is gegeven. Als f gespecificeerd wordt door een lijst van alle deelverzamelingen van X met hun waarden onder f, dan kan de minimale waarde natuurlijk eenvoudig worden gevonden door de hele lijst af te gaan. Maar vaak vereist de declaratie van f minder ruimte, bijvoorbeeld als f is als in de bovenstaande voorbeelden (i) en (ii), in welk geval de vraag naar een polynomiale algorithme minder triviaal wordt. "Polynomiaal" betekent dan: polynomiaal in $|X|$ en $\log B$, waarbij B de grootste teller of noemer van $|f(X')|$ is. Het blijkt dat het probleem van de minimale waarde van een submodulaire functie gereduceerd kan worden tot het scheidingsprobleem voor polyeders van de vorm

(72) $\quad P = \{y \in \mathbb{Q}^X \mid \sum_{x \in X'} y(x) \leq f(X') \text{ voor iedere } X' \subset X\}.$

Bovendien kan het optimaliseringsprobleem voor P eenvoudig binnen polynomiale tijd worden opgelost met de zgn. "greedy" algorithme. Het principe uit §9 geeft ons dan een polynomiale methode om het minimum van f te vinden. Voor een meer gedetailleerde uitwerking wordt verwezen naar [18].

13. GEHEELTALLIGE LINEAIRE PROGRAMMERING

Geheeltallige lineaire programmeringsproblemen vragen om een geheeltallige oplossing voor (1), d.w.z. naar

(73) $\max \{cx \mid Dx \leq b,\ x \text{ geheeltallig}\}$.

Dit "ILP-probleem" blijkt moeilijker dan het LP-probleem zonder geheeltalligheidsvoorwaarden. Zo is het ILP-probleem een NP-volledig probleem, en bestaat er (nog) geen Dualiteitsstelling (een min-max relatie) voor (73). Er is daarom vooralsnog weinig hoop op een polynomiale algorithme. Er bestaan verschillende methoden om (73) te bepalen, zoals "branch-and-bound"-methoden en de methode van de "Gomory-sneden", maar hun looptijd is niet polynomiaal-begrensd (zie Garfinkel en Nemhauser [14]).

De Gomory-sneden-methode kan als volgt worden geformaliseerd. Stel een polyeder P is gegeven. Dan is

(74) $\max \{cx \mid x \in P,\ x \text{ geheeltallig}\} = \max \{cx \mid x \in P_I\}$,

waarbij P_I het convex omhulsel is van de geheeltallige vectoren in P. Omdat het ILP-probleem NP-volledig is, is het optimaliseringsprobleem voor P_I kennelijk ook NP-volledig, gegeven een scheidingsalgorithme voor P. Als we nu P' als volgt definieren:

(75) $P' = \cap H_I$,

waarbij de doorsnede loopt over alle halfruimten H zo dat $P \subset H$. Merk op dat als $H = \{x \mid wx \leq d\}$, waarbij w een geheeltallige vector is waarvan de coördinaten relatief priem zijn, dan is $H_I = \{x \mid wx \leq \lfloor d \rfloor\}$, waarbij $\lfloor d \rfloor$ het gehele deel van d is. H_I ontstaat uit H door H op te schuiven tot het begrenzende hypervlak van H geheeltallige punten bevat.

Eenvoudig is in te zien dat

(76) $P_I \subset P'$,

omdat de inclusie $P \subset H$ impliceert dat $P_I \subset H_I$. Nu kan bewezen worden dat P' weer een polyeder is, en dat

(77) $P_I = P^{(t)}$

voor zekere t, waarbij $P^{(t)} = P^{(t-1)}$'. Dit is de essentie van de Gomory-sneden (zie Gomory [17], Chvátal [3], Schrijver [36]).

Nu zou een antwoord op de volgende vragen een aanzet kunnen zijn tot het toepassen van de ellipsoïde-methode op geheeltallige lineaire programmering. Gegeven een optimaliseringsalgorithme voor het polytoop P, bestaat er een polynomiale scheidingsalgorithme voor P'? (polynomiaal in de looptijd van de optimaliseringsalgorithme voor P). Of is het scheidingsprobleem voor P' NP-volledig? Merk op dat dit in ieder geval een probleem uit de klasse NP is: men kan, gegeven een $y \notin P'$, in polynomiale tijd *bewijzen* dat $y \notin P'$. Hiertoe moet men een halfruimte H geven zo dat $P \subset H$ en $y \notin H_I$. D.w.z., men moet een geheeltallige vector w specificeren zo dat

$$wy > \lfloor \max\{wx \mid x \in P\} \rfloor. \tag{78}$$

Deze ongelijkheid kan in polynomiale tijd worden bewezen met de optimaliseringsalgorithme voor P. Hoe men, gegeven een y, een dergelijke w in polynomiale tijd kan *vinden* is de kern van het scheidingsprobleem voor P'.

14. LITERATUUR

[1] D. BARNETTE, *Path problems and extremal problems for convex polytopes*, in: "Relations between combinatorics and other parts of mathematics", Proc. Symp. Pure Math. 34, Amer. Math. Soc., Providence, R.I., 1979, pp. 25-34.

[2] R.G. BLAND, D. GOLDFARB en M.J. TODD, *The ellipsoid method: a survey*, Tech. Report 476, School of Oper. Res. and Industrial Engineering, Cornell University, Ithaca, N.Y., 1980.

[3] V. CHVÁTAL, *Edmonds polytopes and a hierarchy of combinatorial problems*, Discrete Math. 4 (1973) 305-337.

[4] G.B. DANTZIG, *Maximization of a linear functional of variables subject to linear inequalities*, in: "Activity analysis of production and allocation" (T.C. Koopmans, ed.), J. Wiley, New York, 1951, pp. 339-347.

[5] G.B. DANTZIG, *Linear programming and extensions*, Princeton Univ. Press, Princeton, N.J., 1962.

[6] G.B. DANTZIG, *Expected number of steps of the simplex method for a linear program with convexity constraints*, Tech. Report SOL 80-3, Dept. of Oper. Res., Stanford Univ., Stanford, Ca., 1980.

[7] E.W. DIJKSTRA, *A note on two problems in connexion with graphs*, Numer. Math. 1 (1959) 269-271.

[8] J. EDMONDS, *Maximum matching and a polyhedron with 0,1-vertices*, J. Nat. Bur. Standards Sect. B 69 (1965) 125-130.

[9] L.R. FORD, Jr en D.R. FULKERSON, *Maximum flow through a network*, Canad. J. Math. 8 (1956) 399-404.

[10] D.R. FULKERSON, *Packing rooted directed cuts in a weighted directed graph*, Math. Programming 6 (1974) 1-13.

[11] P. GÁCS en L. LOVÁSZ, *Khachiyan's algorithm for linear programming*, Math. Programming study 14 (1981) 61-68.

[12] D. GALE, H.W. KUHN en A.W. TUCKER, *Linear programming and the theory of games*, in: "Activity analysis of production and allocation" (T.C. Koopmans, ed.), J. Wiley, New York, 1951, pp. 317-329.

[13] M.R. GAREY en D.S. JOHNSON, *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.

[14] R.S. GARFINKEL en G.L. NEMHAUSER, *Integer programming*, J. Wiley, New York, 1972.

[15] D. GOLDFARB en W.Y. SIT, *Worst case behavior of the steepest edge simplex method*, Discrete Applied Math. 1 (1979) 277-285.

[16] D. GOLDFARB en M.J. TODD, *Modifications and implementation of the Shor-Khachian algorithm for linear programming*, Tech. Report 406, School of Oper. Res. and Industrial Engineering, Cornell University, Ithaca, N.Y., 1980.

[17] R.E. GOMORY, *Outline of an algorithm for integer solutions to linear programs*, Bull. Amer. Math. Soc. 64 (1958) 275-278.

[18] M. GRÖTSCHEL, L. LOVÁSZ en A. SCHRIJVER, *The ellipsoid method and its consequences in combinatorial optimization*, verschijnt in Combinatorica.

[19] R. JEROSLOW, *The simplex algorithm with the pivot rule of maximizing criterion improvement*, Discrete Math. 4 (1973) 367-377.

[20] D.B. JUDIN en A.S. NEMIROVSKII, *Informational complexity and effective methods of solution for convex extremal problems*, Ekonomika i Matematicheskie Metody 12 (1976) 357-369 (vertaald: Matekon: Translations of Russian and East European Math. Economics 13 (2) Spring '77, 25-45).

[21] D.B. JUDIN en A.S. NEMIROVSKII, *Optimization methods adapting to the "significant" dimension of the problem*, Automatika i Telemekhanika 38 (1977) No. 4, 75-87 (vertaald: Automation and Remote Control 38 (1977) No 4, 513-524).

[22] R.M. KARP en C.H. PAPADIMITRIOU, *On linear characterizations of combinatorial optimization problems*, Report MIT/LCS/TM-154, Mass. Inst. of Technology, Cambridge, Mass., 1980.

[23] L.G. KHACHIAN, *A polynomial algorithm in linear programming*, Doklady Akademiia Nauk SSSR 244:5 (1979) 1093-1096 (vertaald: Soviet Math. Doklady 20:1 (1979) 191-194).

[24] V. KLEE en G.L. MINTY, *How good is the simplex algorithm?*, in: "Inequalities, III" (O. Shisha, ed.), Academic Press, New York, 1972, pp. 159-175.

[25] G.B. KOLATA, *Mathematicians amazed by Russian discovery*, Science 206 (1979) 545-546.

[26] M.K. KOZLOV, S.P. TARASOV en L.G. KHACHIAN, *Polynomial solvability of convex quadratic programming*, Doklady Akademiia Nauk SSSR **248**:5 (1979) 1049-1050 (vertaald: Soviet Math. Doklady 20:5 (1979) 1108-1111).

[28] J. van LEEUWEN, *Computers en (on-)doenlijke problemen,* dit colloquium.

[29] L. LOVÁSZ, *Normal hypergraphs and the perfect graph conjecture*, Discrete Math. 2 (1972) 253-267.

[30] L. LOVÁSZ, *On the Shannon capacity of a graph*, IEEE Trans. on Information Theory 25 (1979) 1-7.

[31] E.H. McGALL, *A study of the Khachian algorithm for real-world linear programming*, rapport Univac, 1980.

[32] J. von NEUMANN, *On a maximization problem*, manuscript, Institute for Advanced Studies, Princeton, N.J., 1947.

[33] M.W. PADBERG en M.R. RAO, *The Russian method for linear inequalities*, rapport Graduate School of Business Administration, New York University, New York, 1979.

[34] M.W. PADBERG en M.R. RAO, *Minimum cut-sets and* b-*matchings*, rapport Graduate School of Business Administration, New York University, New York, 1979.

[35] M.W. PADBERG en M.R. RAO, *The Russian method and integer programming*, rapport Graduate School of Business Administration, New York University, New York, 1980.

[36] A. SCHRIJVER, *On cutting planes,* Annals of Discrete Math. 9 (1980) 291-296.

[37] N.Z. SHOR, *Generalized gradient methods of nondifferentiable function minimization and their application to problems of mathematical programming*, Ekonomika i Matematicheskie Metody 12 (1976) 337-356.

[38] P. WOLFE, *A bibliography for the ellipsoid method*, rapport IBM Research Center, Yorktown Heights, N.Y., 1980.

EEN GEAUTOMATISEERDE COMPLEXITEITSCLASSIFICATIE VAN COMBINATORISCHE PROBLEMEN

B.J. LAGEWEG

E.L. LAWLER

J.K. LENSTRA

A.H.G. RINNOOY KAN

1. INLEIDING

De afgelopen jaren hebben wij ons intensief beziggehouden met een onderzoek naar de berekeningscomplexiteit van deterministische machinevolgordeproblemen. Het doel was daarbij zo nauwkeurig mogelijk vast te stellen waar de grillige grens loopt tussen "gemakkelijke" en "moeilijke" probleemtypen. We noemen een probleem *gemakkelijk* als het oplosbaar is in polynomiale tijd en *moeilijk* als het "NP-hard" is.

De door ons onderzochte klasse bevat 4536 combinatorische optimaliseringsproblemen. Het werd al spoedig een saai en tijdrovend karwei de behaalde complexiteitsresultaten met de hand bij te houden. Voor het verrichten van deze werkzaamheden ontwikkelden wij daarom het computerprogramma MSPCLASS. De invoer van dit programma bestaat uit twee lijsten van problemen waarvan bekend is dat zij gemakkelijk resp. moeilijk zijn. Met gebruikmaking van een bepaalde *partiële ordening* die is gedefinieerd op de klasse problemen gaat het programma na wat de status van ieder probleem is (*gemakkelijk*, *open* of *moeilijk*) en bepaalt het vier deelklassen van *maximaal gemakkelijke*, *minimaal open*, *maximaal open* en *minimaal moeilijke* problemen. Deze uitvoer is van groot nut gebleken voor ons onderzoek.

In §2 wordt het programma MSPCLASS in algemene termen beschreven. In §3 definiëren we een klasse van 120 één-machineproblemen, en in §4 demonstreren

we de toepassing van het programma op deze beperkte klasse. (De stand van zaken voor de volledige klasse staat beschreven in [11]; een voorlopig verslag over een klasse van 4158 problemen is te vinden in [10].) In §5 worden mogelijke verfijningen en uitbreidingen naar andere gebieden van onderzoek gesuggereerd.

Het zou nuttig zijn wanneer een programma als MSPCLASS tevens uitsluitsel zou verschaffen over het minimale aantal te behalen onderzoeksresultaten dat alle resterende open problemen doet verdwijnen. In de Appendix wordt aangetoond dat de bepaling van dit aantal op zich een moeilijk probleem is.

2. HET PROGRAMMA MSPCLASS

Wij veronderstellen dat de lezer vertrouwd is met de basisbegrippen uit de complexiteitstheorie en dat een bespreking van hun definities en eigenschappen achterwege kan blijven. Voor een uitstekende inleiding zij verwezen naar [5].

Het programma MSPCLASS is bestemd voor een welgedefinieerde klasse S van probleemtypen. Voor elk tweetal problemen $P,P' \in S$ is het programma in staat vast te stellen of al dan niet geldt dat $P \to P'$, waarbij $\to$ een gegeven *partiële ordening* is met de volgende eigenschappen:

(i) als $P \to P'$ en P' is gemakkelijk, dan is P gemakkelijk;

(ii) als $P \to P'$ en P is moeilijk, dan is P' moeilijk.

De relatie $P \to P'$ geldt bijvoorbeeld als de verzameling instanties van P evident bevat is in de verzameling instanties van P'. In het algemeen impliceert $P \to P'$ dat P *reduceerbaar* is tot P'.

De invoer van het programma bestaat uit een klasse I^* van problemen waarvan bekend is dat zij gemakkelijk zijn en een klasse $I^!$ van problemen waarvan bekend is dat zij moeilijk zijn. Het programma splitst de klasse S vervolgens in drie klassen van gemakkelijke, moeilijke en open problemen:

$$S^* = \{P \in S \mid \exists\, P' \in I^* : P \to P'\},$$

$$S^! = \{P \in S \mid \exists\, P' \in I^! : P' \to P\},$$

$$S^? = S - (S^* \cup S^!).$$

(De klassen S^* en $S^!$ zijn natuurlijk disjunct; zo niet, dan is er een fout gemaakt of een buitengewoon verrassend resultaat behaald.) Het programma bepaalt bovendien vier deelklassen van problemen die *minimaal* of *maximaal* zijn t.a.v. de relatie $\to$:

$$S^*_{max} = \{P \in S^* \mid \neg \exists\ P' \in S^* - \{P\}\colon P \to P'\},$$
$$S^?_{min} = \{P \in S^? \mid \neg \exists\ P' \in S^? - \{P\}\colon P' \to P\},$$
$$S^?_{max} = \{P \in S^? \mid \neg \exists\ P' \in S^? - \{P\}\colon P \to P'\},$$
$$S^!_{min} = \{P \in S^! \mid \neg \exists\ P' \in S^! - \{P\}\colon P' \to P\}.$$

De uitvoer van het programma omvat een telling van het aantal problemen in S^*, $S^?$ en $S^!$ alsmede een volledige opsomming van de problemen in S^*_{max}, $S^?_{min}$, $S^?_{max}$ en $S^!_{min}$.

De omvang van S^*, $S^?$ en $S^!$ biedt een weerspiegeling van de voortgang van het onderzoek. De problemen in S^*_{max} en $S^!_{min}$ representeren in zekere zin de *essentiële resultaten*, waaruit m.b.v. de relatie $\to$ alle andere resultaten zijn af te leiden; de minimale invoer wordt dan ook gegeven door $I^* = S^*_{max}$ en $I^! = S^!_{min}$. De problemen in $S^?_{min}$ en $S^?_{max}$ zijn voor de hand liggende objecten voor verder onderzoek.

N.B. Voor de definitie van S^*, $S^?$ en $S^!$ is het niet noodzakelijk dat de relatie $\to$ een partiële ordening is. Voor de definitie van S^*_{max}, $S^?_{min}$, $S^?_{max}$ en $S^!_{min}$ is dit echter wel van essentieel belang. Het is namelijk heel goed mogelijk dat bepaalde deelverzamelingen van equivalente complexiteit zijn t.a.v. de relatie $\to$; zij corresponderen met sterk samenhangende componenten van de gerichte graaf met knopenverzameling S en kanten gedefinieerd door $\to$. Alle problemen in zo'n equivalentieklasse kunnen bijvoorbeeld maximaal gemakkelijk zijn, terwijl toch geen enkel behoort tot S^*_{max}, zoals hierboven gedefinieerd.

3. EEN KLASSE ÉÉN-MACHINEPROBLEMEN

Gezien de omvang en de ingewikkeldheid van de door ons bestudeerde klasse machinevolgordeproblemen geven wij er de voorkeur aan de werking van het programma MSPCLASS te demonstreren aan de hand van een beperkte klasse van 120 één-machineproblemen. De nochtans noodzakelijke definities en toelichtingen volgen hieronder.

In het algemeen vraagt iedere instantie van een één-machineprobleem om het vinden van een optimaal toegelaten *schedule* voor een verzameling van n *opdrachten* J_j (j = 1,...,n). Een schedule bestaat uit een verzameling disjuncte tijdsintervallen en een aanduiding van de opdracht die in elk interval door de machine wordt bewerkt. De *toelaatbaarheid* van een schedule wordt bepaald

door een aantal gespecificeerde parameters en voorwaarden. De *optimaliteit* wordt beoordeeld m.b.t. een gegeven doelstellingsfunctie.

Meer in het bijzonder wordt in iedere probleeminstantie de volgende informatie expliciet of impliciet vastgelegd.

(1) Voor elke opdracht J_j ($j = 1,\ldots,n$) is een niet-negatieve geheeltallige *bewerkingsduur* p_j gegeven. Teneinde toegelaten te zijn, moet een schedule p_j tijdseenheden toewijzen aan elke J_j. Voor ieder schedule definiëren we S_j en C_j als de *aanvangstijd* resp. de *voltooiingstijd* van J_j, d.w.z. het tijdstip waarop J_j voor het eerst (het laatst) in bewerking is.

(2) Als het schedule *niet-preëmptief* dient te zijn, dan moet elke J_j zonder onderbreking worden bewerkt van S_j tot $C_j = S_j+p_j$. Als het schedule daarentegen *preëmptief* mag zijn, dan mag de bewerking van een opdracht willekeurig vaak worden onderbroken.

(3) Voor elke J_j is een niet-negatieve geheeltallige *beschikbaarheidstijd* r_j gegeven. In een toegelaten schedule moet gelden dat $r_j \leq S_j$ voor elke J_j.

(4) *Volgorderelaties* tussen de opdrachten zijn gegeven in de vorm van een acyclische gerichte graaf $G = (\{1,\ldots,n\},A)$. In een toegelaten schedule moet gelden dat $C_j \leq S_k$ als $(j,k) \in A$.

(5) M.b.t. de doelstellingsfunctie kan voor elke J_j een niet-negatieve geheeltallige *aflevertijd* d_j en een geheeltallig *gewicht* w_j gegeven zijn. Voor ieder schedule definiëren we $L_j = C_j-d_j$ als de *laatheid* van J_j, $T_j = \max\{0,C_j-d_j\}$ als de *traagheid*, en $U_j = 0$ als $C_j \leq d_j$, $U_j = 1$ als $C_j > d_j$ als de *boeteëenheid*.

De te minimaliseren *doelstellingsfunctie* is nu één van de volgende acht types:

- de *maximale voltooiingstijd* $C_{max} = \max_j\{C_j\}$;
- de *maximale laatheid* $L_{max} = \max_j\{L_j\}$;
- de *totale voltooiingstijd* $\sum_j C_j$;
- de *totale gewogen voltooiingstijd* $\sum_j w_j C_j$;
- de *totale traagheid* $\sum_j T_j$;
- de *totale gewogen traagheid* $\sum_j w_j T_j$;
- het *aantal te late opdrachten* $\sum_j U_j$;
- het *gewogen aantal te late opdrachten* $\sum_j w_j U_j$.

In deze klasse machinevolgordeproblemen wordt ieder probleemtype gedefinieerd door een sextupel $(\pi_0,\pi_1,\pi_2,\pi_3,\pi_4,\pi_5)$, waarbij elke component een eigenschap aanduidt die wordt gedeeld door alle probleeminstanties. De component π_0 verschaft informatie over de *machines*, de componenten π_1,π_2,π_3,π_4 geven karakteristieken van de *opdrachten* weer, en de component π_5 betreft de *doelstellingsfunctie*. Deze componenten kunnen de volgende waarden aannemen, waarbij $\circ$ het lege symbool voorstelt.

(0) $\pi_0 = 1$: één machine.

(1) $\pi_1 \in \{p_j=1, \circ\}$, met

$\pi_1 = p_j=1$: $p_j = 1$ $(j = 1,\ldots,n)$;

$\pi_1 = \circ$: elke p_j is een willekeurig niet-negatief geheel getal.

(2) $\pi_2 \in \{\circ, pmtn\}$, met

$\pi_2 = \circ$: preëmptie is niet toegestaan;

$\pi_2 = pmtn$: preëmptie is toegestaan.

(3) $\pi_3 \in \{\circ, r_j\}$, met

$\pi_3 = \circ$: $r_j = 0$ $(j = 1,\ldots,n)$;

$\pi_3 = r_j$: elke r_j is een willekeurig niet-negatief geheel getal.

(4) $\pi_4 \in \{\circ, tree, prec\}$, met

$\pi_4 = \circ$: G bevat geen kanten (de opdrachten zijn onafhankelijk);

$\pi_4 = tree$: G heeft hetzij ingraad ten hoogste gelijk aan één voor elk punt hetzij uitgraad ten hoogste gelijk aan één voor elk punt;

$\pi_4 = prec$: G is een willekeurige acyclische gerichte graaf.

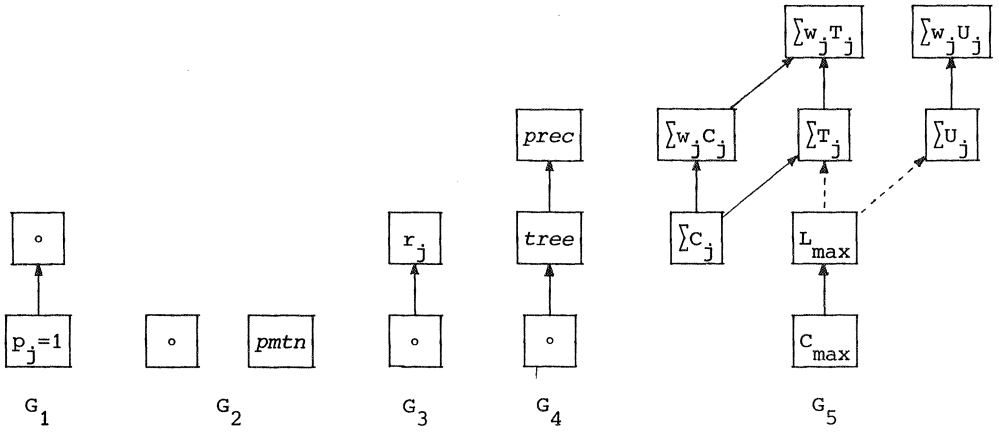
(5) $\pi_5 \in \{C_{max}, L_{max}, \sum C_j, \sum w_j C_j, \sum T_j, \sum w_j T_j, \sum U_j, \sum w_j U_j\}$.

Er zijn zodoende $1\times2\times2\times2\times3\times8 = 192$ probleemtypen. Zij worden geschreven in de vorm $1|\pi_4,\pi_3,\pi_2,\pi_1|\pi_5$ (vgl. [6]).

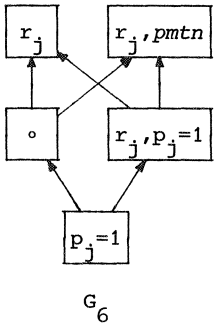
Sommige eigenschappen gedefinieerd door de componenten van het sextupel zijn eenvoudige generalisaties van andere. Zo is *prec* een generalisatie van *tree* en zijn $\sum w_j C_j$ en $\sum T_j$ generalisaties van $\sum C_j$. De constructie van de relatie $\rightarrow$ is gebaseerd op deze observatie.

De mogelijke waarden van de component π_i corresponderen met de punten van de gerichte graaf G_i weergegeven in Figuur 1 $(i = 1,\ldots,5)$. Er loopt een kant van π naar π' als π' een directe generalisatie is van π. Voor twee problemen $P = 1|\pi_4,\pi_3,\pi_2,\pi_1|\pi_5$ en $P' = 1|\pi'_4,\pi'_3,\pi'_2,\pi'_1|\pi'_5$ hebben we $P \rightarrow P'$ als hetzij $\pi_i = \pi'_i$ hetzij G een gericht pad bevat van π_i naar π'_i, voor $i = 1,\ldots,5$.

De relatie $\rightarrow$ kan worden uitgebreid door rekening te houden met verbanden tussen probleemtypen die subtieler zijn dan eenvoudige generalisaties. De onderbroken kanten van L_{max} naar $\sum T_j$ en $\sum U_j$ kunnen worden toegevoegd op grond van het volgende argument. Voor iedere instantie van een probleem met doelstellingsfunctie L_{max} geldt dat de minimale waarde van L_{max} gelijk is aan de minimale waarde van L waarvoor er een schedule bestaat met $\sum T'_j = \sum U'_j = 0$, waarbij T'_j en U'_j worden gedefinieerd t.a.v. aflevertijden $d'_j = d_j + L$. De minimale L kan worden gevonden door middel van een "bisection search", bestaande uit een polynomiaal begrensd aantal toepassingen van een algoritme die in staat is het probleem met $\sum T_j$ of $\sum U_j$ i.p.v. L_{max} op te lossen. Hieruit volgt



Figuur 1. De grafen G_i (i = 1,...,5).



Figuur 2. De graaf G_6.

de (Turing) reduceerbaarheid van ieder L_{max} probleem tot de overeenkomstige $\sum T_j$ en $\sum U_j$ problemen.

Een verdere uitbreiding van de relatie $\rightarrow$ is gebaseerd op enige overwegingen m.b.t. preëmptie. Voor één-machineproblemen kan eenvoudig worden aangetoond dat er geen reden is de bewerking van een opdracht op een niet-geheeltallig tijdstip de onderbreken en dat preëmptie geen voordeel oplevert als alle beschikbaarheidstijden gelijk zijn. Hieruit volgt dat, als $\pi_1 = p_j=1$ of $\pi_3 = \circ$, wij mogen aannemen dat $\pi_2 = \circ$. Er blijven nu slechts vijf relevante combinaties van waarden over voor de componenten π_1, π_2 en π_3. Zij corresponderen met de punten π_6 van de gerichte graaf G_6 in Figuur 2, waarvan de kanten worden geïmpliceerd door G_1, G_2 en G_3. Het totale aantal te onderscheiden probleemtypen is nu teruggebracht van 192 tot 120; zij worden geschreven in de vorm $1|\pi_4,\pi_6|\pi_5$.

4. TOEPASSING VAN MSPCLASS OP DE KLASSE ÉÉN-MACHINEPROBLEMEN

Het programma MSPCLASS is geïmplementeerd in PASCAL op de Control Data Cyber 170-750 van SARA. Hieronder demonstreren wij de toepassing van het programma op de in §3 gedefinieerde klasse één-machineproblemen.

Wij merken op dat het programma tweemaal wordt gedraaid, waarbij de invoer verschillende veronderstellingen weerspiegelt omtrent de codering van probleeminstanties die numerieke gegevens bevatten. Onder een standaard *binaire* codering geldt dat de gemakkelijke problemen oplosbaar zijn in *strikt polynomiale* tijd en dat de moeilijke problemen *NP-hard* zijn in de *normale* betekenis van het woord. Onder een *unaire* codering geldt dat de gemakkelijke problemen oplosbaar zijn in *pseudopolynomiale* tijd en dat de moeilijke problemen *NP-hard* zijn in de *sterke* betekenis van het woord (vgl. [4;18]).

De meest recente uitvoer is weergegeven in Figuren 3 en 4. De lijsten bevatten nuttige informatie zoals literatuurverwijzingen waar de betreffende resultaten te vinden zijn, een aanduiding van algemene algoritmische technieken waarmee sommige gemakkelijke problemen kunnen worden opgelost, het symbool # voor open problemen die zowel minimaal als maximaal zijn, en afkortingen van geschikte vertrekpunten voor transformaties in het geval van moeilijke problemen.

Na combinatie van de resultaten voor binaire en unaire coderingen ontstaan zes klassen problemen:

- unair en binair gemakkelijk (51 problemen);
- unair en binair open (4 problemen, alle met het $\sum T_j$ criterium);
- unair en binair moeilijk (62 problemen);
- unair gemakkelijk, binair open (1 probleem: $1||\sum T_j$);
- unair gemakkelijk, binair moeilijk (2 problemen: $1||\sum w_j U_j$ en $1|r_j, pmtn|\sum w_j U_j$);
- unair open, binair moeilijk (0 problemen).

Bij deze loven wij aantrekkelijke prijzen uit voor degene die als eerste één van de open problemen oplost: een fles Californische champagne voor een polynomiale algoritme en een Edammer kaas voor een transformatie die moeilijkheid bewijst.

5. CONCLUSIES

Het programma MSPCLASS is van groot nut gebleken bij het vastleggen van de

SINGLE MACHINE SCHEDULING, BINARY ENCODING

DATE: 81:03:05

NUMBER OF PROBLEMS	MINIMAL	TOTAL	MAXIMAL
EASY		51	8
OPEN	2	5	3
HARD	12	64	
TOTAL		120	

MAXIMAL EASY PROBLEMS

1/RJ,PJ=1/SUMWJTJ	LAGEWEG (TRANSPORTATION PROBLEM)
1/RJ,PJ=1/SUMWJUJ	LAGEWEG (TRANSPORTATION PROBLEM)
1/RJ,PMTN/SUMCJ	BAKER 1974
1/RJ,PMTN/SUMUJ	LAWLER 1981 (DYNAMIC PROGRAMMING)
1/TREE/SUMWJCJ	HORN 1972; SIDNEY 1975
1/PREC,RJ,PJ=1/SUMCJ	LAWLER: COFFMAN GRAHAM 1972
1/PREC,RJ,PMTN/LMAX	BLAZEWICZ 1976
1/PREC,RJ/CMAX	LAWLER 1973

MINIMAL OPEN PROBLEMS	MAXIMAL OPEN PROBLEMS
1//SUMTJ	1/RJ,PMTN/SUMTJ
1/TREE,PJ=1/SUMTJ	1/TREE,RJ,PJ=1/SUMTJ
	1/TREE/SUMTJ

MINIMAL HARD PROBLEMS

1//SUMWJTJ	3PT	LAWLER 1977; LENSTRA RINNOOY KAN BRUCKER 1977
1//SUMWJUJ	KS	KARP 1972
1/RJ,PMTN/SUMWJCJ	3PT	LABETOULLE LAWLER LENSTRA RINNOOY KAN 1979
1/RJ/LMAX	3PT	LENSTRA RINNOOY KAN BRUCKER 1977
1/RJ/SUMCJ	3PT	LENSTRA RINNOOY KAN BRUCKER 1977
1/TREE,PJ=1/SUMWJTJ	3PT	LENSTRA RINNOOY KAN 1980
1/TREE,PJ=1/SUMUJ	S3P	LENSTRA RINNOOY KAN 1980
1/TREE,RJ,PJ=1/SUMWJCJ	3PT	LENSTRA RINNOOY KAN 1980
1/TREE,RJ,PMTN/SUMCJ	3PT	LENSTRA 1980
1/PREC,PJ=1/SUMWJCJ	LA	LAWLER 1978; LENSTRA RINNOOY KAN 1978
1/PREC,PJ=1/SUMTJ	CL	LENSTRA RINNOOY KAN 1978
1/PREC/SUMCJ	LA	LAWLER 1978; LENSTRA RINNOOY KAN 1978

Figuur 3.

stand van zaken in de theorie van de machinevolgordeproblemen, een gebied waarop de laatste jaren aanzienlijke vorderingen zijn gemaakt. Het programma heeft goede diensten bewezen bij het interpreteren van nieuwe resultaten en heeft vaak de richting aangegeven van verdere onderzoeksinspanningen.

Het systeem kan nog op vele manieren worden verfijnd, bijvoorbeeld door een nadere uitwerking van de relatie $\rightarrow$ zoals gesuggereerd in §2 en door een verdere analyse van de klasse open problemen. Zo zou het prettig zijn als het

SINGLE MACHINE SCHEDULING, UNARY ENCODING

DATE: 81:03:05

NUMBER OF PROBLEMS	MINIMAL	TOTAL	MAXIMAL
EASY		54	8
OPEN	2	4	3
HARD	11	62	
TOTAL		120	

MAXIMAL EASY PROBLEMS

1/RJ,PJ=1/SUMWJTJ	LAGEWEG (TRANSPORTATION PROBLEM)
1//SUMTJ	LAWLER 1977
1/RJ,PMTN/SUMCJ	BAKER 1974
1/RJ,PMTN/SUMWJUJ	LAWLER 1981 (DYNAMIC PROGRAMMING)
1/TREE/SUMWJCJ	HORN 1972; SIDNEY 1975
1/PREC,RJ,PJ=1/SUMCJ	LAWLER: COFFMAN GRAHAM 1972
1/PREC,RJ,PMTN/LMAX	BLAZEWICZ 1976
1/PREC,RJ/CMAX	LAWLER 1973

MINIMAL OPEN PROBLEMS		MAXIMAL OPEN PROBLEMS
1/RJ,PMTN/SUMTJ	#	1/RJ,PMTN/SUMTJ
1/TREE,PJ=1/SUMTJ		1/TREE,RJ,PJ=1/SUMTJ
		1/TREE/SUMTJ

MINIMAL HARD PROBLEMS

1//SUMWJTJ	3PT	LAWLER 1977; LENSTRA RINNOOY KAN BRUCKER 1977
1/RJ,PMTN/SUMWJCJ	3PT	LABETOULLE LAWLER LENSTRA RINNOOY KAN 1979
1/RJ/LMAX	3PT	LENSTRA RINNOOY KAN BRUCKER 1977
1/RJ/SUMCJ	3PT	LENSTRA RINNOOY KAN BRUCKER 1977
1/TREE,PJ=1/SUMWJTJ	3PT	LENSTRA RINNOOY KAN 1980
1/TREE,PJ=1/SUMUJ	S3P	LENSTRA RINNOOY KAN 1980
1/TREE,RJ,PJ=1/SUMWJCJ	3PT	LENSTRA RINNOOY KAN 1980
1/TREE,RJ,PMTN/SUMCJ	3PT	LENSTRA 1980
1/PREC,PJ=1/SUMWJCJ	LA	LAWLER 1978; LENSTRA RINNOOY KAN 1978
1/PREC,PJ=1/SUMTJ	CL	LENSTRA RINNOOY KAN 1978
1/PREC/SUMCJ	LA	LAWLER 1978; LENSTRA RINNOOY KAN 1978

Figuur 4.

programma het minimale aantal te behalen onderzoeksresultaten zou berekenen dat alle resterende open problemen doet verdwijnen. Deze verfijning is waarschijnlijk lastig te implementeren, aangezien het probleem op zich moeilijk is, zoals we in de Appendix aantonen. Dit complexiteitsresultaat weerspiegelt de practische onmogelijkheid de minimale kosten van het voltooien van een onderzoeksproject te bepalen.

Zelfs zonder dergelijke verfijningen geloven wij dat een uitbreiding van

onze aanpak naar andere klassen combinatorische optimaliseringsproblemen een vergelijkbaar rendement zou kunnen opleveren. Routering van voertuigen, locatie en allocatie, en het ontwerpen van netwerken zijn gebieden die in aanmerking lijken te komen.

6. APPENDIX. DE BEREKENING VAN EEN MINIMAAL VOLLEDIG ONDERZOEKSPROGRAMMA

Stel dat de stand van zaken wordt weergegeven door een transitieve acyclische gerichte graaf $G = (S,\rightarrow)$ waarvan elke knoop het label *gemakkelijk*, *open* of *moeilijk* draagt, met inachtneming van de volgende eisen:
(i) als $P \rightarrow P'$ en P' is *gemakkelijk*, dan is P *gemakkelijk*, en
(ii) als $P' \rightarrow P$ en P' is *moeilijk*, dan is P *moeilijk*.
Een *volledig onderzoeksprogramma* (VOP) voor G wordt gedefinieerd door een deelverzameling van *open* knopen met de eigenschap dat het mogelijk is het label van elk van deze knopen te veranderen (in *gemakkelijk* of *moeilijk*) zodanig dat door toepassing van (i) en (ii) elke andere *open* knoop ook van label verandert. Wij zijn geïnteresseerd in het berekenen van een VOP van minimale omvang.

Op het eerste gezicht lijkt dit probleem nauw verwant aan een bekend gemakkelijk probleem. Als we vragen naar een minimaal aantal *open* knopen die het label *opgelost* dienen te krijgen opdat er voor elke andere *open* knoop P een *opgeloste* knoop P' bestaat met $P \rightarrow P'$ of $P' \rightarrow P$, dan zou toepassing van de stelling van Menger op de transitieve kern van G in polynomiale tijd het antwoord geven. We vragen echter naar een minimaal aantal *open* knopen die het label *gemakkelijk* of *moeilijk* moeten krijgen zodanig dat er voor elke andere *open* knoop P hetzij een *gemakkelijke* knoop P' is met $P \rightarrow P'$ hetzij een *moeilijke* knoop P' is met $P' \rightarrow P$. Wij zullen aantonen dat dit probleem moeilijk is.

Het bewijs geschiedt door middel van een transformatie vanuit het volgende NP-volledige probleem:

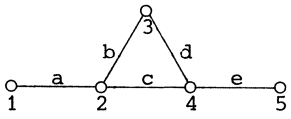
> PUNTOVERDEKKING [5,[GT1]]: Gegeven een graaf $G = (V,E)$ en een geheel getal k, bevat V een deelverzameling U van ten hoogste k punten zodanig dat elke kant in E incident is met tenminste één punt in U?

Zij $\ell = |V|+k$. Gegeven een instantie van PUNTOVERDEKKING construeren we een gerichte graaf $G = (S,\rightarrow)$ op de volgende wijze (vgl. Figuur 5):

- voor elk punt $v \in V$ zijn er $\ell+2$ knopen $v_1,\ldots,v_\ell,\bar{v},\bar{\bar{v}} \in S$ en $\ell+1$ kanten $v_1 \rightarrow \bar{v}$, ..., $v_\ell \rightarrow \bar{v}$, $\bar{v} \rightarrow \bar{\bar{v}}$;

PUNTOVERDEKKING

Instantie: G = (V,E):

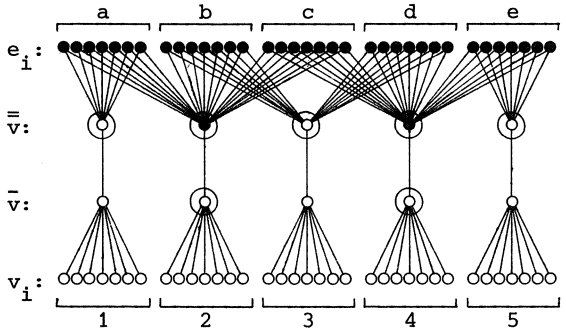


k = 2.

Oplossing: U = {2,4}.

VOP PROBLEEM

Instantie: $G = (S,\rightarrow)$ (alle kanten zijn naar boven gericht; kanten die volgen uit de transitiviteit zijn weggelaten):



$\ell = 7$.

Oplossing: VOP = {◎,◉}; ○:*gemakkelijk*; ●: *moeilijk*.

Figuur 5. Illustratie van de transformatie.

- voor elke kant e = {v,w} $\in$ E zijn er ℓ knopen $e_1,\ldots,e_\ell \in S$ en 2ℓ kanten $\overline{\overline{v}} \rightarrow e_1, \ldots, \overline{\overline{v}} \rightarrow e_\ell, \overline{\overline{w}} \rightarrow e_1, \ldots, \overline{\overline{w}} \rightarrow e_\ell$;
- alle andere kanten volgen uit de transitiviteit;
- alle knopen dragen het label *open*.

Wij beweren dat PUNTOVERDEKKING een oplossing heeft dan en slechts dan als er een VOP is van ten hoogste ℓ knopen.

Onderstel dat G een puntoverdekking $U \subseteq V$ bevat van grootte ten hoogste k. Een VOP van grootte ten hoogste ℓ wordt dan verkregen door elke $\overline{\overline{v}}$ ($v \in U$) het label *moeilijk* te geven en elke $\overline{\overline{v}}$ ($v \in V-U$) en elke $\bar{v}$ ($v \in U$) het label *gemakkelijk*; door toepassing van (i) en (ii) krijgt elke e_i ($i = 1,\ldots,\ell$; $e \in E$) het label *moeilijk* en elke $\bar{v}$ ($v \in V-U$) en v_i ($i = 1,\ldots,\ell$; $v \in V$) het label *gemakkelijk*.

Onderstel nu, omgekeerd, dat G een VOP $\subset S$ bevat van grootte ten hoogste ℓ. Uit de keuze van ℓ volgt onmiddellijk dat de knopen v_i en e_i ($i = 1,\ldots,\ell$)

alleen het label *gemakkelijk* resp. *moeilijk* kunnen krijgen, en bovendien dat geen van deze knopen behoort tot het VOP. Voor elke $v \in V$ bevat het VOP één knoop uit $\{\bar{v},\bar{\bar{v}}\}$ of beide. In het eerste geval kan dat alleen $\bar{\bar{v}}$ zijn die dan het label *gemakkelijk* draagt. In het tweede geval draagt $\bar{v}$ het label *gemakkelijk* en $\bar{\bar{v}}$ het label *moeilijk*. De *moeilijke* knopen $\bar{\bar{v}} \in S$, waarvan er niet meer dan $\ell-|V| = k$ kunnen zijn, stellen ons in staat alle e_i $(i = 1,\ldots,\ell;\ e \in E)$ ook het label *moeilijk* te geven. Hieruit volgt dat de overeenkomstige punten $v \in V$ een puntoverdekking van grootte ten hoogste k vormen. *Q.e.d.*

7. VERANTWOORDING

De auteurs zijn R.M. Karp erkentelijk voor zijn stimulerende aanwezigheid bij het concipiëren van dit project. Dit onderzoek werd gedeeltelijk gesteund door NATO Special Research Grant 9.2.02 (SRG.7) en door NSF Grant MCS78-20054.

8. LITERATUUR

1. K.R. BAKER (1974) *Introduction to Sequencing and Scheduling*, Wiley, New York.
2. J. BLAZEWICZ (1976) Scheduling dependent tasks with different arrival times to meet deadlines. In: E. GELENBE (ed.) (1976) *Modelling and Performance Evaluation of Computer Systems*, North-Holland, Amsterdam, 57-65.
3. E.G. COFFMAN, JR., R.L. GRAHAM (1972) Optimal scheduling for two-processor systems. *Acta Informat.* 1,200-213.
4. M.R. GAREY, D.S. JOHNSON (1978) "Strong" NP-completeness results: motivation, examples and implications. *J. Assoc. Comput. Mach.* 25,499-508.
5. M.R. GAREY, D.S. JOHNSON (1979) *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, San Francisco.
6. R.L. GRAHAM, E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discrete Math.* 5,287-326.
7. W.A. HORN (1972) Single-machine job sequencing with treelike precedence ordering and linear delay penalties. *SIAM J. Appl. Math.* 23,189-202.
8. R.M. KARP (1972) Reducibility among combinatorial problems. In: R.E. MILLER, J.W. THATCHER (eds.) (1972) *Complexity of Computer Computations*, Plenum Press, New York, 85-103.

9. J. LABETOULLE, E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN (1979) Preemptive scheduling of uniform machines subject to release dates. Report BW 99, Mathematisch Centrum, Amsterdam.

10. B.J. LAGEWEG, E.L. LAWLER, J.K. LENSTRA (1976) Machine scheduling problems: computations, complexity and classification; in honour of A.H.G. Rinnooy Kan upon the occasion of the defense of his doctoral thesis, January 28, 1976. Report BN 30, Mathematisch Centrum, Amsterdam (out of print).

11. B.J. LAGEWEG, E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN (1981) Computer aided complexity classification of deterministic scheduling problems. Report BW 138, Mathematisch Centrum, Amsterdam.

12. E.L. LAWLER (1973) Optimal sequencing of a single machine subject to precedence constraints. *Management Sci.* 19,544-546.

13. E.L. LAWLER (1977) A "pseudopolynomial" algorithm for sequencing jobs to minimize total tardiness. *Ann. Discrete Math.* 1,331-342.

14. E.L. LAWLER (1978) Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Ann. Discrete Math.* 2,75-90.

15. E.L. LAWLER (1981) Unpublished result.

16. J.K. LENSTRA (1980) Unpublished result.

17. J.K. LENSTRA, A.H.G. RINNOOY KAN (1978) Complexity of scheduling under precedence constraints. *Oper. Res.* 26,22-35.

18. J.K. LENSTRA, A.H.G. RINNOOY KAN (1979) Computational complexity of discrete optimization problems. *Ann. Discrete Math.* 4,121-140.

19. J.K. LENSTRA, A.H.G. RINNOOY KAN (1980) Complexity results for scheduling chains on a single machine. *European J. Oper. Res.* 4,270-275.

20. J.K. LENSTRA, A.H.G. RINNOOY KAN, P. BRUCKER (1977) Complexity of machine scheduling problems. *Ann. Discrete Math.* 1,343-362.

21. J.B. SIDNEY (1975) Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Oper. Res.* 23,283-298.

UITGAVEN IN DE SERIE MC SYLLABUS

Onderstaande uitgaven zijn verkrijgbaar bij het Mathematisch Centrum, Kruislaan 413, 1098 SJ Amsterdam, tel.020-5929333

MCS 1.1 F. GÖBEL & J. VAN DE LUNE, *Leergang Besliskunde, deel 1: Wiskundige basiskennis*, 1965. ISBN 90 6196 014 2.

MCS 1.2 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 2: Kansberekening*, 1965. ISBN 90 6196 015 0.

MCS 1.3 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 3: Statistiek*, 1966. ISBN 90 6196 016 9.

MCS 1.4 G. DE LEVE & W. MOLENAAR, *Leergang Besliskunde, deel 4: Markovketens en wachttijden*, 1966. ISBN 90 6196 017 7.

MCS 1.5 J. KRIENS & G. DE LEVE, *Leergang Besliskunde, deel 5: Inleiding tot de mathematische besliskunde*, 1966. ISBN 90 6196 018 5.

MCS 1.6a B. DORHOUT & J. KRIENS, *Leergang Besliskunde, deel 6a: Wiskundige programmering 1*, 1968. ISBN 90 6196 032 0.

MCS 1.6b B. DORHOUT, J. KRIENS & J.TH. VAN LIESHOUT, *Leergang Besliskunde, deel 6b: Wiskundige programmering 2*, 1977. ISBN 90 6196 150 5.

MCS 1.7a G. DE LEVE, *Leergang Besliskunde, deel 7a: Dynamische programmering 1*, 1968. ISBN 90 6196 033 9.

MCS 1.7b G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7b: Dynamische programmering 2*, 1970. ISBN 90 6196 055 X.

MCS 1.7c G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7c: Dynamische programmering 3*, 1971. ISBN 90 6196 066 5.

MCS 1.8 J. KRIENS, F. GÖBEL & W. MOLENAAR, *Leergang Besliskunde, deel 8: Minimaxmethode, netwerkplanning, simulatie*, 1968. ISBN 90 6196 034 7.

MCS 2.1 G.J.R. FÖRCH, P.J. VAN DER HOUWEN & R.P. VAN DE RIET, *Colloquium Stabiliteit van differentieschema's, deel 1*, 1967. ISBN 90 6196 023 1.

MCS 2.2 L. DEKKER, T.J. DEKKER, P.J. VAN DER HOUWEN & M.N. SPIJKER, *Colloquium Stabiliteit van differentieschema's, deel 2*, 1968. ISBN 90 6196 035 5.

MCS 3.1 H.A. LAUWERIER, *Randwaardeproblemen, deel 1*, 1967. ISBN 90 6196 024 X.

MCS 3.2 H.A. LAUWERIER, *Randwaardeproblemen, deel 2*, 1968. ISBN 90 6196 036 3.

MCS 3.3 H.A. LAUWERIER, *Randwaardeproblemen, deel 3*, 1968. ISBN 90 6196 043 6.

MCS 4 H.A. LAUWERIER, *Representaties van groepen*, 1968. ISBN 90 6196 037 1.

MCS 5 J.H. VAN LINT, J.J. SEIDEL & P.C. BAAYEN, *Colloquium Discrete wiskunde*, 1968.
ISBN 90 6196 044 4.

MCS 6 K.K. KOKSMA, *Cursus ALGOL 60*, 1969. ISBN 90 6196 045 2.

MCS 7.1 *Colloquium Moderne rekenmachines, deel 1*, 1969. ISBN 90 6196 046 0.

MCS 7.2 *Colloquium Moderne rekenmachines, deel 2*, 1969. ISBN 90 6196 047 9.

MCS 8 H. BAVINCK & J. GRASMAN, *Relaxatietrillingen*, 1969.
ISBN 90 6196 056 8.

MCS 9.1 T.M.T. COOLEN, G.J.R. FÖRCH, E.M. DE JAGER & H.G.J. PIJLS, *Elliptische differentiaalvergelijkingen, deel 1*, 1970.
ISBN 90 6196 048 7.

MCS 9.2 W.P. VAN DEN BRINK, T.M.T. COOLEN, B. DIJKHUIS, P.P.N. DE GROEN, P.J. VAN DER HOUWEN, E.M. DE JAGER, N.M. TEMME & R.J. DE VOGELAERE, *Colloquium Elliptische differentiaalvergelijkingen, deel 2*, 1970.
ISBN 90 6196 049 5.

MCS 10 J. FABIUS & W.R. VAN ZWET, *Grondbegrippen van de waarschijnlijkheidsrekening*, 1970. ISBN 90 6196 057 6.

MCS 11 H. BART, M.A. KAASHOEK, H.G.J. PIJLS, W.J. DE SCHIPPER & J. DE VRIES, *Colloquium Halfalgebra's en positieve operatoren*, 1971.
ISBN 90 6196 067 3.

MCS 12 T.J. DEKKER, *Numerieke algebra*, 1971. ISBN 90 6196 068 1.

MCS 13 F.E.J. KRUSEMAN ARETZ, *Programmeren voor rekenautomaten; De MC ALGOL 60 vertaler voor de EL X8*, 1971. ISBN 90 6196 069 X.

MCS 14 H. BAVINCK, W. GAUTSCHI & G.M. WILLEMS, *Colloquium Approximatietheorie*, 1971. ISBN 90 6196 070 3.

MCS 15.1 T.J. DEKKER, P.W. HEMKER & P.J. VAN DER HOUWEN, *Colloquium Stijve differentiaalvergelijkingen, deel 1*, 1972. ISBN 90 6196 078 9.

MCS 15.2 P.A. BEENTJES, K. DEKKER, H.C. HEMKER, S.P.N. VAN KAMPEN & G.M. WILLEMS, *Colloquium Stijve differentiaalvergelijkingen, deel 2*, 1973. ISBN 90 6196 079 7.

MCS 15.3 P.A. BEENTJES, K. DEKKER, P.W. HEMKER & M. VAN VELDHUIZEN, *Colloquium Stijve differentiaalvergelijkingen, deel 3*, 1975.
ISBN 90 6196 118 1.

MCS 16.1 L. GEURTS, *Cursus Programmeren, deel 1: De elementen van het programmeren*, 1973. ISBN 90 6196 080 0.

MCS 16.2 L. GEURTS, *Cursus Programmeren, deel 2: De programmeertaal ALGOL 60*, 1973. ISBN 90 6196 087 8.

MCS 17.1 P.S. STOBBE, *Lineaire algebra, deel 1*, 1974. ISBN 90 6196 090 8.

MCS 17.2 P.S. STOBBE, *Lineaire algebra, deel 2*, 1974. ISBN 90 6196 091 6.

MCS 17.3 N.M. TEMME, *Lineaire algebra , deel 3*, 1976. ISBN 90 6196 123 8.

MCS 18 F. VAN DER BLIJ, H. FREUDENTHAL, J.J. DE IONGH, J.J. SEIDEL & A. VAN WIJNGAARDEN, *Een kwart eeuw wiskunde 1946-1971, Syllabus van de Vakantiecursus 1971*, 1974. ISBN 90 6196 092 4.

MCS 19 A. HORDIJK, R. POTHARST & J.Th. RUNNENBURG, *Optimaal stoppen van Markovketens*, 1974. ISBN 90 6196 093 2.

MCS 20 T.M.T. COOLEN, P.W. HEMKER, P.J. VAN DER HOUWEN & E. SLAGT, *ALGOL 60 procedures voor begin- en randwaardeproblemen*, 1976. ISBN 90 6196 094 0.

MCS 21 J.W. DE BAKKER (red.), *Colloquium Programmacorrectheid*, 1975. ISBN 90 6196 103 3.

MCS 22 R. HELMERS, F.H. RUYMGAART, M.C.A. VAN ZUYLEN & J. OOSTERHOFF, *Asymptotische methoden in de toetsingstheorie; Toepassingen van naburigheid*, 1976. ISBN 90 6196 104 1.

MCS 23.1 J.W. DE ROEVER (red.), *Colloquium Onderwerpen uit de biomathematica, deel 1*, 1976. ISBN 90 6196 105 X.

MCS 23.2 J.W. DE ROEVER (red.), *Colloquium Onderwerpen uit de biomathematica, deel 2*, 1976. ISBN 90 6196 115 7.

MCS 24.1 P.J. VAN DER HOUWEN, *Numerieke integratie van differentiaalvergelijkingen, deel 1: Eenstapsmethoden*, 1974. ISBN 90 6196 106 8.

MCS 25 *Colloquium Structuur van programmeertalen*, 1976. ISBN 90 6196 116 5.

MCS 26.1 N.M. TEMME (ed.), *Nonlinear analysis, volume 1*, 1976. ISBN 90 6196 117 3.

MCS 26.2 N.M. TEMME (ed.), *Nonlinear analysis, volume 2*, 1976. ISBN 90 6196 121 1.

MCS 27 M. BAKKER, P.W. HEMKER, P.J. VAN DER HOUWEN, S.J. POLAK & M. VAN VELDHUIZEN, *Colloquium Discretiseringsmethoden*, 1976. ISBN 90 6196 124 6.

MCS 28 O. DIEKMANN, N.M. TEMME (EDS), *Nonlinear Diffusion Problems*, 1976. ISBN 90 6196 126 2.

MCS 29.1 J.C.P. BUS (red.), *Colloquium Numerieke programmatuur, deel 1A, deel 1B*, 1976. ISBN 90 6196 128 9.

MCS 29.2 H.J.J. TE RIELE (red.), *Colloquium Numerieke programmatuur, deel 2*, 1976. ISBN 90 6196 144 0

MCS 30

MCS 31 J.H. VAN LINT (red.), *Inleiding in de coderingstheorie*, 1976. ISBN 90 6196 136 X.

MCS 32 L. GEURTS (red.), *Colloquium Bedrijfssystemen*, 1976. ISBN 90 6196 137 8.

MCS 33 P.J. VAN DER HOUWEN, *Differentieschema's voor de berekening van waterstanden in zeeën en rivieren*, 1977. ISBN 90 6196 138 6.

MCS 34 J. HEMELRIJK, *Oriënterende cursus mathematische statistiek*, ISBN 90 6196 139 4.

MCS 35 P.J.W. TEN HAGEN (red.), *Colloquium Computer Graphics*, 1977. ISBN 90 6196 142 4.

MCS 36 J.M. AARTS, J. DE VRIES, *Colloquium Topologische Dynamische Systemen*, 1977. ISBN 90 6196 143 2.

MCS 37 J.C. van Vliet (red.), *Colloquium Capita Datastructuren*, 1978. ISBN 90 6196 159 9.

MCS 38.1 T.H. KOORNWINDER (ED.), *Representations of locally compact groups with applications*, 1979. ISBN 90 6196 161 0.

MCS 38.2 T.H. KOORNWINDER (ED.), *Representations of locally compact groups with applications*, 1979. ISBN 90 6196 181 5.

MCS 39 O.J. VRIEZE & G.L. WANROOIJ, *Colloquium Stochastische Spelen*, 1979. ISBN 90 6196 167 X.

MCS 40 J. VAN TIEL, *Convexe Analyse*, 1979. ISBN 90 6196 187 4.

MCS 41 H.J.J. TE RIELE (ED.), *Colloquium Numerical Treatment of Integral Equations*, 1979. ISBN 90 6196 189 0.

MCS 42 J.C. VAN VLIET (RED.), *Colloquium Capita Implementatie van Programmeertalen*, 1980. ISBN 90 6196 191 2.

MCS 43 A.M. COHEN & H.A. WILBRINK, *Eindige groepen (Een inleidende cursus)*, 1980. ISBN 90 6196 203 X.

MCS 44 J.G. VERWER (ED.), *Numerical solution of partial differential equations*, 1980. ISBN 90 6196 205 6.

MCS 45 P. KLINT (RED.), *Colloquium hogere programmeertalen en computerarchitectuur*, 1980. ISBN 90 6196 206 4.

MCS 46.1 P.M.G. APERS (RED.), *Colloquium Databankorganisatie*, 1981. ISBN 90 6196 212 9.

MCS 46.2 P.M.G. APERS (RED.), *Colloquium Databankorganisatie*, 1981. ISBN 90 6196 232 3.

MCS 47.1 P.W. HEMKER (ED.), *NUMAL, numerical procedures in ALGOL 60*, Part I: *General information and indices*, 1981. ISBN 90 6196 217 X.

MCS 47.2 P.W. HEMKER (ED.), *NUMAL, numerical procedures in ALGOL 60*, Part II: *Elementary procedures, algebraic evaluations*, 1981. ISBN 90 6196 217 X.

MCS 47.3 P.W. HEMKER (ED.), *NUMAL, numerical procedures in ALGOL 60*, Part III: *Linear algebra, part I*, 1981. ISBN 90 6196 217 X.

MCS 47.4 P.W. HEMKER (ED.), *NUMAL, numerical procedures in ALGOL 60*, Part IV: *Linear algebra, part II*, 1981. ISBN 90 6196 217 X.

MCS 47.5 P.W. HEMKER (ED.), *NUMAL, numerical procedures in ALGOL 60*, Part V: *Analytical evaluations, analytical problems, part I*, 1981. ISBN 90 6196 217 X.

MCS 47.6 P.W. HEMKER (ED.), *NUMAL, numerical procedures in ALGOL 60*, Part VI: *Analytical problems, part II*, 1981. ISBN 90 6196 217 X.

MCS 47.7 - P.W. HEMKER (ED.), *NUMAL, numerical procedures in ALGOL 60*, Part VII: *Special functions and constants, interpolation and approximation*, 1981. ISBN 90 6196 217 X.

MCS 48.1 P.M.B. VITANYI, J. VAN LEEUWEN & P. VAN EMBE BOAS (RED.), *Colloquium complexiteit en algoritmen, deel 1*, 1982. ISBN 90 6196 237 4

MCS 48.2 P.M.B. VITANYI, J. VAN LEEUWEN & P. VAN EMDE BOAS (RED.), *Colloquium complexiteit en algoritmen, deel 2*, 1982. ISBN 90 6196 246 3

MCS 49 T.H. KOORNWINDER (ED.), *The structure of real semisimple Lie groups*, 1982. ISBN 90 6196 239 0.

MCS 50 H. NIJMEIJER, *Inleiding systeemtheorie*, 1982. ISBN 90 6196 240 4.