

Statistische programmatuur en computers: enkele ontwikkelingen

C. van de Wijngaart (Technisch Centrum FSW, Universiteit van Amsterdam)

In Nederland is, zoals in verschillende andere Europese landen, aan het eind van de zestiger jaren de eerste statistische programmatuur voor een grotere kring van gebruikers beschikbaar gekomen, met name voor het soort gebruikers dat niet zelf programmeert. De achterstand die in die periode bestond op de beschikbaarheid van programmatuur in de Verenigde Staten is in de laatste tien jaar geheel ingelopen. Bij grotere concentraties van gebruikers in Nederland, zoals aan de universiteiten, is tegenwoordig vrijwel dezelfde programmatuur beschikbaar als in de V.S. Met name kan men hierbij denken aan algemene pakketten als BMDP, PSTAT en SPSS, en meer op één type analyse gerichte programmatuur zoals CLUSTAN en MULTIVARIANCE. Tevens blijkt in Europa ontwikkelde programmatuur thans voor Amerikaanse gebruikers interessant te worden, en niet zelden daar in gebruik. De belangrijkste oorzaken voor de achterstand waren vooral gelegen in de beschikbare apparatuur. (Misschien mag zelfs wel gezegd worden dat programmeren in Europa bepaald niet het probleem was; zie bijvoorbeeld de ontwikkeling van Algol 60.) Nu in Nederland de modernste computers beschikbaar zijn en men intussen voldoende ervaring met zulke apparatuur heeft, beschikt de gebruiker in het algemeen over de beste programmatuur die er bestaat. Of deze relatieve kwaliteit ook aan de feitelijke eisen voldoet, is de vraag.

Bij de beantwoording van deze vraag dienen meerdere aspecten onderscheiden te worden.

Een zeer belangrijk aspect is de kwaliteit van de statistische methode die geprogrammeerd is. Dit onderwerp blijft in het volgende buiten beschouwing; het is een statistisch probleem dat buiten de verantwoordelijkheid van de programmeur ligt.

Bij een tweede aspect, de kwaliteit van de gebruikte algorithmen (de vertaling van de statistische methode in "rekenopdrachten") dragen programmeur én opdrachtgevende statisticus gezamenlijk de verantwoordelijkheid. In de praktijk blijkt men al te vaak geheel op de programmeur te vertrouwen, en gaat ook de gebruiker er té gemakkelijk vanuit dat het geprogrammeerde algoritme een vlekkeloze weergave van de methode is.

Naast "deugdelijke methoden" en "correcte uitkomsten" blijven er nog vele eisen over, die aan de programmatuur gesteld worden. Deze liggen vooral op het terrein van de "hanteerbaarheid" van de programmatuur; de realisatie ervan berust grotendeels bij de ontwerper/programmeur. De programmatuur dient

bijvoorbeeld "efficient" te zijn, d.w.z. zo weinig mogelijk geheugencapaciteit en rekentijd te vergen. Minimum eis is dat de programmatuur, bij de beoogde omvang van te verwerken data etc., in de beschikbare configuratie "past", maar steeds vaker óók dat het geheugenbeslag beneden de limiet blijft die interactief gebruik toestaat.

Een andere voorziening, minder noodzakelijk en meer bedoeld om gebruikers werk te besparen, is de mogelijkheid van gebruik van een system file of een andere faciliteit om eenmaal ingevoerde (of berekende) data voor volgende bewerkingen te bewaren, tezamen met gegevens als labels en missing values.

Veel in de laatste jaren ontwikkelde programmatuur voldoet echter aan hogere eisen van "gebruikersvriendelijkheid". Om er enkele te noemen:

- bundeling van verwante programmatuur in pakketten en bibliotheken,
- standaardisatie van in- en uitvoer,
- gebruik van stuurtalen (met keywords),
- eisen aan de gebruikersdocumentatie,
- beschikbaarheid van testdata en voorbeelduitvoer.

Aan sommige andere eisen van dit type wordt soms minder goed of helemaal niet voldaan:

- begrijpelijke foutmeldingen,
- gebruikersvriendelijke behandeling van invoerfouten,
- layout, lettertype en printkwaliteit van de uitvoer (bijv. op A4-formaat, geschikt voor overname in publicaties),
- grafische uitvoer.

Veel van deze onderwerpen vindt men terug bij het nader beschouwen van de driedeling in statistische programmatuur die zich de laatste jaren aftekent:

1. Algemene pakketten, zoals BMDP, PSTAT en SPSS. De belangrijkste kenmerken zijn: gebruik van een stuurtaal en system file, een grote range van analyse methoden maar óók data manipulatie-technieken, en uitvoerige documentatie.
2. Speciale pakketten zoals MULTIVARIANCE en CLUSTAN. De methoden liggen op één gebied, en zijn op zich soms interessanter dan die in de algemene pakketten. Gebruikersvriendelijke voorzieningen zoals stuurtaal, system file en documentatie zijn niet altijd even optimaal.
3. Losse programma's. Het aantal losse programma's overtreft verre het aantal via pakketten beschikbare programma's, terwijl het gebruik van pakketten juist veel hoger is. Losse programma's zijn voor een deel ongetwijfeld terecht "nooit in een pakket opgenomen", n.l. verouderd, niet algemeen genoeg, methodologisch zwak, etc. Voor een belangrijk deel echter geldt juist het omgekeerde, en zijn deze programma's juist de meest interessante,



de nieuwste. Dat pakketten dan op die ontwikkeling achterlopen wordt veroorzaakt door de grote investeringen die nodig zijn om losse programma's (of liever: algorithmen) om te bouwen naar de vele eisen die een pakket stelt.

Bij een toenemend aantal losse programma's wordt er blijkbaar rekening mee gehouden dat opname in een pakket niet of niet tijdig plaats vindt: de programma's hebben zélf voorzieningen als een stuurtaal, wat dan de opname in pakketten extra bemoeilijkt.

De documentatie van losse programma's is zeer gevarieerd.

Goede gebruikershandleidingen, zowel bij pakketten als losse programma's, bevatten tegenwoordig een methodologische inleiding op de gebruikte methode(n), een beschrijving van het programma en van de besturing daarvan, technische gegevens, en voorbeelden van setup en uitvoer.

Een aantal criteria voor "goede programmatuur" is voor de uiteindelijke gebruiker minder interessant en in hoofdzaak van belang voor de programmeur en zijn collega's. We noemen hier:

- de overdraagbaarheid van de programmatuur naar andere configuraties, andere merken apparatuur etc. Bijvoorbeeld speelt hierbij het gebruik van algemene, gestandaardiseerde programmeertalen een rol.

Hoewel in het algemeen aan de portabiliteit van software nog heel wat mankeert kan opgemerkt worden dat meer en meer software tegenwoordig wordt gekocht, geleased of gehuurd, in plaats van op verzoek van de toekomstige gebruiker (of door hem zelf) geschreven.

- De kosten van ontwikkeling, testen en onderhoud. Onderwerpen die hierbij aan de orde komen zijn o.a.: hoe wordt het ontwerp ingericht zodat zo weinig mogelijk ontwikkelingstijd nodig is, welke eisen stelt men aan de structuur van het programma, welke programmeertaal wordt gebruikt, en hoe dient de "interne" documentatie van het programma te zijn. Hierbij enkele opmerkingen: Omvangrijke, moderne programmatuur is veelal buitengewoon gecompliceerd, wat de ontwikkelingskosten zéér hoog maakt. Terwijl de prijzen van hardware momenteel snel omlaag gaan, gaan die van software juist snel omhoog. Gemiddeld genomen, maar zeer afhankelijk van de kwaliteit van de software, zijn echter de kosten van het uittesten, documenteren en vooral het onderhouden van de software nog (veel) hoger dan die van de ontwikkeling ervan.

Een laatste, vooral voor gebruikers belangrijke, faciliteit in statistische (en andere) programmatuur is de interactieve bruikbaarheid.

Hoewel interactieve terminals (en dus ook de benodigde "systeem software") in Europa al zo'n tien jaar in gebruik zijn, zijn er naar verhouding nog maar weinig écht interactieve programma's, waarbij dus niet slechts de setup "conversationeel" wordt ingevoerd, maar de gebruiker vooral het programma kan besturen op grond van tussenresultaten. Uiteraard is dat niet voor iedere statistische analyse zinvol. Beschouwt men echter ook andere toepassingen, zoals de uitvoering van meerdere van elkaar afhankelijke analyses, de acquisitie van data (bijv. enquêtering), correctie van ruwe data, of statistisch onderwijs, dan wordt het aantal mogelijke toepassingen van interactieve programmatuur bijzonder groot. De beschikbaarheid van interactieve programmatuur loopt dus niet in de pas met die van de bijbehorende hardware. Mogelijke oorzaken zijn o.a.:

- Gebruikers van statistische programmatuur zijn wellicht niet voldoende bekend met de mogelijkheden.
  - Standaard programmatuur richt zich op "iedere gebruiker". Een groot percentage gebruikers beschikte enige jaren geleden echter nog niet over interactieve terminals; die beschikbaarheid lijkt nu pas algemeen te worden. O.a. zal dit het gevolg zijn van de prijsdalingen van hardware.
  - De organisatie van interactieve programmatuur wijkt nogal af van die van batch programmatuur, waardoor batch programma's meestal niet om te bouwen zijn en herschreven moeten worden.
  - Wil men, bij de besturing van interactieve programmatuur, iets verder gaan dan het intypen van door het programma gevraagde aantallen, optie-nummers etc., met name in de richting van computerreacties op ingetypte tekst, "vriendelijke" reacties op typefouten e.d., dan blijkt direct dat enigszins intelligent gebruik van (vooral natuurlijke) taal nog grote problemen geeft, en zeer complexe, soepele en toch stabiele programmatuur vereist.
- Wat men op dit moment hierdoor veelvuldig ziet gebeuren is dat de terminal de ponskaart vervangt, maar dat de programmatuur "niet bij" is: de gebruiker typt zijn batch-setup op een terminal in, en krijgt na enige tijd uitvoer, meestal op een (centraal opgestelde) regeldrukker (omdat het batch programma te veel uitvoer levert, met te lange regels). Het enige voordeel is dan dat men zijn job niet aan de balie hoeft in te leveren.

De koppeling van andere hardware, zoals graphic displays, aan interactieve programma's is in het algemeen in standaardprogrammatuur nog helemaal niet gerealiseerd. (Ook hier bestaan echter uitzonderingen.) Bij graphic displays bijvoorbeeld spelen hogere kosten en belangrijk grotere afhankelijkheid van de standaard software een rol.



Het achterblijven van de toepassingsprogrammatuur op de technische ontwikkelingen beperkt zich helaas niet tot het gebruik van interactieve terminals, maar is een veel algemener verschijnsel, met belangrijke consequenties.

De belangrijkste hardware ontwikkeling van de laatste jaren is waarschijnlijk wel die van de micro processor. De gevolgen hiervan zijn goed te demonstrenen aan de hand van de (zak-)rekenmachine. Rond 1965 bestonden er elektrische rekenmachines, waarmee "met maar éénmaal intypen van waarnemingsuitkomsten  $x$  en  $y$ " (voor vrij kleine getallen) men tegelijk de sommen, de kwadraatsommen en het kruisproduct van  $x$  en  $y$  kon bepalen. Dit waren mechanisch zeer gecompliceerde machines, die enkele duizenden guldens kostten. Omstreeks 1970 kwamen hiervoor in de plaats elektronische rekenmachines die, met dezelfde invoer, ook de gemiddelden, standaarddeviaties en product-moment correlatie berekenden. Deze machines waren ongeveer even duur als hun voorgangers, maar hadden minder last van storingen, en waren snel en geruisloos. Rond 1975 verscheen de pocket calculator, voorzien van micro processor, vrijwel meteen in diverse uitvoeringen, zoals voor financiële, statistische, technische, programmeurs- en scholieren-doeleinden.

De hardware programma's (chips) blijken steeds weer complexer gemaakt te kunnen worden, de geheugens groter, zodat er nog wel méér ontwikkeling te verwachten is. Een aantal pocket calculators is bovendien programmeerbaar. Deze apparaten zijn véél goedkoper dan hun voorgangers (een eenvoudige uitvoering is al voor ca. f 10,- verkrijgbaar), het is een massaproduct geworden, er is felle concurrentie, er zijn tientallen merken en nog veel meer types op de markt.

In feite heeft de pocket calculator, door zijn betrekkelijke eenvoud en de enorme omzet die ermee te bereiken is, model gestaan voor vele andere toepassingen van de micro processor, waarvan in dit verband vooral van belang zijn de micro computer, de terminal (uitgebreid met een toenemende hoeveelheid "locale intelligentie" en geheugen) en de toepassingen in andere computerapparatuur.

Tegelijk ontstonden bovendien andere technische vernieuwingen op computer gebied, zoals nieuwe geheugentechnieken, floppy disks en cassette tapes. We kunnen ons, zonder op technische details in te gaan, afvragen welke hardware ontwikkelingen van belang zijn voor de ontwikkeling, onderhoud en gebruik van statistische programmatuur.

Op dit moment onderscheidt men ruwweg drie categorieën computers: "grote" computers, zoals men die bijvoorbeeld aantreft in universitaire rekencentra,

"mini computers", in wetenschappelijke omgevingen vaak te vinden in laboratoria, en micro computers, zo hier en daar op iemands bureau.

De grote computer is zeer kostbaar, is meestal voorzien van zeer uiteenlopende randapparatuur en wordt voor sterk uiteenlopende doeleinden gebruikt. Mini computers hebben een geringere capaciteit, zijn beduidend goedkoper, en worden vaak voor één type verwerking gebruikt. De "miniaturisering" begint hier voelbaar te worden: terwijl de capaciteit toeneemt daalt de prijs. Dit maakt de mini computer interessant voor betrekkelijk omvangrijk werk, dat tot nu toe op grote computers werd uitgevoerd, vooral als daar vele anderen gebruik van maken (problemen van prioriteit of privacy) of de specifieke toepassing een bepaald ander merk/type vergt (bijv. bij tekstverwerking, administratie, computer aided instruction). Ook bij nieuwe automatiseringsprojecten zal men meer dan vroeger aan een "eigen" mini denken.

De situatie m.b.t. micro computers laat zich het best omschrijven als "chaotisch". Er bestaan tientallen "grote" merken en enkele honderden verschillende types, die zéér uiteenlopend kunnen zijn. Ook de mogelijkheden tot uitbreiding en de prijzen variëren sterk, al is een algemeen kenmerk dat de prijzen laag zijn. Een micro computer is nauwelijks anders te definiëren dan een "door één of enkele chips gestuurde computer", zodat we ons hier beperken tot die apparatuur, die kant en klaar te koop is, door de gebruiker geprogrammeerd kan worden en zelfstandig in staat is "rekenprogramma's" uit te voeren.

Dan blijft er nog een heleboel over, van micro's voor huishoudelijk gebruik tot en met micro's met kleuren graphics-uitvoer van wiskundige functies. Een privé computer van enkele duizenden guldens mag dan heel aantrekkelijk lijken, soms kan men zich afvragen of het meer is dan speelgoed.

De geschiktheid van een computer voor bepaalde toepassingen hangt, evenals die van programmatuur, van een reeks factoren af. De kwaliteit van computers is zelden ter discussie, de onderlinge technische verschillen des te meer. Wie bijvoorbeeld matrices van  $50 \times 50$  wil inverteren heeft beslist een snelle machine nodig, met zo groot mogelijke woordlengte en een forse geheugen-capaciteit. Voor de berekening van een eenvoudige regressie-analyse daarentegen kan men beduidend lagere machine-eisen stellen, en wellicht volstaan met een bescheiden micro. Voor statistische toepassingen echter strandt men al gauw op de kleine woordlengte van micro computers, heel vaak 8-bits. Ook eisen aan in- en uitvoer en randapparatuur spelen een rol.

Andere aspecten zijn, evenals bij programmatuur, minder op de toepasbaarheid gericht maar wél erg belangrijk, zoals de kosten, het onderhoud, de service.



De gebruiker van statistische programma's dient met name "de programmatuur" als criterium te hanteren, als hij computers vergelijkt.

De eerste vraag die zich dan voordoet is: welke programma's worden er door de leverancier bijgeleverd? Voor de meeste leveranciers is die vraag nogal pijnlijk: zeker de leveranciers van hardware raken, wat toepassingsprogrammatuur betreft, steeds verder achter. Fabrikanten van (middel)grote computers hebben veelal geen of weinig interessante applicatieprogramma's te bieden, maar men wijst wel graag op wat gebruikers ter beschikking stellen (al dan niet gratis), en dat is als regel inderdaad véél. Leveranciers van micro computers hebben 't moeilijker. Door de decentrale locatie van micro's hebben gebruikers vrijwel geen contact, zodat er geen gebruikersbibliotheken zijn. Bovendien is de apparatuur nieuw, zodat er nergens, zoals bij grote computers, al jaren aan programmatuur gewerkt is. Soms levert de fabrikant daarom zélf maar iets, zoals op statistisch terrein bijv. programmaatjes voor enkelvoudige regressie, tijdreeksen, correlaties en variantie analyse.

Interessant hierbij is dat de fabrikant dit soort programma's steeds meer door software houses laat ontwikkelen, omdat men zelf de risico's (onvoorspelbare levertijden en investeringen) liever niet loopt.

Een tweede methode om aan programmatuur te komen, zonder die zelf te ontwikkelen, is overname van derden. Bij grote computers heeft men een redelijke kans op succes (het gebeurt ook meer en meer), vooral als men binnen één merk blijft. Er zijn n.l. relatief niet veel grote computers, én niet veel merken grote computers; op "de eigen machine" vindt men meestal al heel wat, en overige interessante adressen zijn wel bekend, zodat het zoeken niet al te moeilijk is. Blijft men niet binnen één computermerk dan komt het portabiliteitsprobleem om de hoek kijken.

Enige overname van toepassingsprogramma's voor micro's kan men, gezien de plaatsing van micro's en het aantal programmeurs per apparaat, pas verwachten als programma-producenten zich enigszins zouden organiseren; daar ziet het nog niet naar uit, zodat dit soort overname niet eenvoudig zal zijn. Tenslotte kan men zélf programma's (laten) ontwikkelen. Voor statistische programmatuur op grotere computers begint dit steeds minder te gebeuren, wat vooral veroorzaakt wordt door de hoge kosten (denk bijvoorbeeld aan zeer bewerkelijke interactieve programma's), en in mindere mate door de afnemende vraag.

Natuurlijke speelt ook de toename van het huren en kopen van programma's hierbij een rol.

Micro computers zal men, zoals we zagen, voorlopig vooral zélf moeten programmeren, hoe kostbaar dat ook is. De lage aanschafprijs van micro's komt hierdoor wel in een geheel ander daglicht. Helaas doet zich hierbij nog

een ander probleem voor, n.l. dat van programmeertalen. De verwende gebruiker van grote installaties zal ontdekken dat het, wat dit betreft, bij micro's maar behelpen is: behalve in hun eigen assembleertaal zijn de meeste micro's slechts in BASIC, of een andere taal van dit niveau, programmeerbaar. Dat is een flinke stap terug! Maar zelfs als het apparaat een taal als Pascal aankan, en het programmeren minder vervelend wordt, blijkt dat micro's in zo'n taal beperkingen stellen, die de overdraagbaarheid van de programma's bemoeilijken. "Eigen uitbreidingen" van zulke talen komen ook voor, maar die zijn, evenals op grote computers, meestal meer bezwaarlijk dan nuttig (ook vanwege de overdraagbaarheid).

Er is nog een geheel andere ontwikkeling op dit terrein vermeldenswaard. De chip-techniek is al zó ver gevorderd dat men erin slaagt om zeer gecompliceerde programma's, zoals bijv. een LISP-processor, in een chip te verwerken. Deze ontwikkeling van hardware programma's is buitengewoon kostbaar, zodat op dit terrein waarschijnlijk alleen productie te verwachten is bij een flinke omzetverwachting. Het is niet ondenkbaar dat enkele standaardtechnieken die status zullen bereiken. Voor statistische pakketten, al zijn het soms best-sellers, is dit nog niet te voorzien; beschikbaarheid op floppy disks (óók t.b.v. micro's) ligt eerder in de lijn.

Een conclusie zal zijn dat de ontwikkeling van micro computers, technisch en financieel gezien, zeer interessant is, en ongetwijfeld vergaande consequenties zal hebben, óók voor statistische programmatuur. Op dit moment echter zal voor het meeste statistische gebruik de overgang naar een micro computer eerder een stap terug zijn, en een kostbare.

Er is echter een andere manier om micro's te gebruiken, die waarschijnlijk veel zinvoller is: de koppeling van micro's aan een groter systeem.

Ten eerste kan daarmee de micro als intelligente terminal gebruikt worden, zodat de programma's van het grote systeem beschikbaar komen. Vervolgens kan men, als daar voorzieningen voor aanwezig zijn, lokaal eenvoudige vóór- en nabewerkingen uitvoeren, zoals editing, en zal men daarbij minder afhankelijk zijn van storingen van de centrale apparatuur. Nú echter heeft de ontwikkeling van programmatuur voor de micro óók meer zin, mits men althans een programmeertaal gebruikt die de grote machine óók kent: die programma's zijn dan zowel op de eigen micro's, als op de centrale apparatuur, als op andere aangesloten micro's bruikbaar. De laatste zouden dan liefst van het zelfde type moeten zijn. Vergaande standaardisatie zal voorlopig echter hoogstens binnen één organisatie mogelijk zijn.

Er is, niet in de laatste plaats, ook nog een menselijke factor. De meest talentvolle programmeurs blijken vaak wat bezeten van hun vak, verrukt van



nieuwe technische ontwikkelingen, en dol op een "eigen" micro. De kans op technisch fraaie, maar niet overdraagbare, of zelfs leuke maar zinloze, programma's wordt zodoende voor micro's wat groter. (Micro computers lijken daarom o.a. erg geschikt om in het onderwijs te worden ingeschakeld.)

#### Conclusie en samenvatting.

Statistische programmatuur én computers zijn momenteel in drie categorieën in te delen:

- groot (duur, traditioneel, bedoeld voor "alles", snel),
- middelgroot (minder duur, flink in ontwikkeling, vaak voor specifieke doeleinden gebruikt),
- klein (goedkoop, vaak zéér nieuw, zeer uiteenlopend).

De ontwikkeling van programmatuur blijft echter meer en meer achter bij die van apparatuur, die in een stroomversnelling is geraakt. Een ander belangrijk verschil is dat apparatuur, gemeten aan de prestaties, steeds goedkoper wordt, en programmatuur steeds duurder. In veel planningen wordt er al vanuit gegaan dat de kosten van software die van hardware zullen overschrijden.

De eisen die men aan statistische programmatuur stelt zijn, naast methodologische en algorithmische, de laatste jaren vooral die van gebruikersvriendelijkheid. Aan verschillende van deze eisen wordt in de grote, dure, software voldaan. Interactieve bruikbaarheid is in het algemeen nog nauwelijks aanwezig; blijktbaar is het werk te omvangrijk, het ontwikkelingstempo te laag, de kosten te hoog.

Dit verschijnsel van achterstand op de techniek ziet men nog duidelijker bij de nieuwste ontwikkeling: de micro computer. De goedkope micro computer zal, bij onverstandig gebruik, de achterstand van statistische software op hardware vergroten en de kosten van software ontwikkeling nog verder opdrijven. Dit negatieve effect kan vermeden worden door binnen de eigen organisatie streng te standaardiseren en koppeling aan de centrale machine verplicht te stellen; buiten de eigen organisatie dient, met name via de oprichting van gebruikersorganisaties, de uitwisseling van programmatuur bevorderd te worden.

KWANTITATIEVE METHODEN IN NEDERLAND

De Publicatiecommissie van de SOR heeft het initiatief genomen om in KM een serie artikelen te laten verschijnen gewijd aan groepen in Nederland die zich met de beoefening van Kwantitatieve Methoden bezighouden. Het doel van de serie is tweeledig: in ruime kring bekend maken wie er in Nederland aan kwantitatieve methoden doen en het bevorderen van contacten tussen vakgenoten. De redactie is verheugd over dit initiatief en wil er graag ruimte voor beschikbaar stellen. Voor bijdragen tot deze reeks zal de Publicatiecommissie van de SOR contact zoeken met in aanmerking komende groepen. Spontane inzendingen zijn echter ook van harte welkom (te richten aan: Leonard Fortuin, N.V. Philips' Gloeilampenfabrieken, CQM, Gebouw VN-711, 5600 MD EINDHOVEN, telefoon 040-735941).

Redactie KM.

KWANTITATIEVE METHODEN BIJ ... nr. 2



De Technische Hogeschool Eindhoven

C.B. Tilanus<sup>\*</sup>

Een beknopt overzicht van onderwijs en onderwijsgebonden onderzoek in de Operations Research bij de Technische Hogeschool Eindhoven, en een beschrijving van een concreet, doorsnee afstudeerproject.

1. Inleiding

Bij de Technische Hogeschool Eindhoven (THE) moeten we drie zaken onderscheiden: (1) research, (2) operations research, en (3) onderwijs in de operations research. Over (2) de operations research (OR) moeten we het hebben. Bij instellingen voor wetenschappelijk onderwijs zoals de THE wordt wel (1) research bedreven die misschien ooit van pas kan komen bij de OR, bijvoorbeeld research in de wiskunde of de informatica, maar daar hebben we het niet over. Er wordt ook (3) onderwijs gegeven in de OR, maar daar hebben we het ook niet over.

Naast het onderwijs in de OR wordt de OR ook zelf beoefend. Dit gebeurt op twee plaatsen bij de THE: bij de Afdeling der Bedrijfskunde en bij de Onderafdeling der Wiskunde. De vakgroepen waar OR beoefend wordt heten:

---

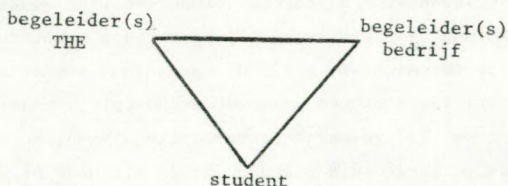
\*Technische Hogeschool Eindhoven, Afdeling der Bedrijfskunde, Postbus 513, 5600 MB Eindhoven



- bij Wiskunde: vakgroep Kansrekening, Statistiek, Operations Research en Systeemtheorie; 27 man sterk;
- bij Bedrijfskunde: vakgroep Operations Research en Statistiek; 9 man sterk;
- vakgroep Kwantitatieve Aspecten van Beheersingssystemen; 12 man sterk.

Het merendeel van de OR betreft zogenaamd onderwijsgebonden onderzoek. Dat wil zeggen, ieder die onderwijs geeft over een bepaald onderwerp, krijgt er automatisch tijd bij om onderzoek over dat onderwerp te doen teneinde de kwaliteit van zijn onderwijs op peil te houden. (Of dit onderzoek resultaten oplevert, wordt niet gecontroleerd en valt ook moeilijk te controleren; toch wil men in de toekomst meer onderzoek via aparte "geldstromen" financieren en controleren.) In paragraaf 2 geef ik een overzicht van het gegeven onderwijs, met het oog op het eraan gebonden onderzoek.

Een afstudeerproject in de OR in een bedrijf vind ik een ideale vorm om OR te beoefenen aan een Technische Hogeschool. In Eindhoven studeren per jaar ongeveer 70 bedrijfskundige ingenieurs af, waarvan 20 in de OR, benevens 25 wiskundige ingenieurs, waarvan 10 in de OR. Een afstudeerproject duurt in de regel 9 maanden. In deze periode moet de aanstaande ingenieur door middel van zelfstandig onderzoek het bewijs van praktische bekwaamheid leveren. Als het goed is, ontstaat de volgende driehoeksverhouding:



De student heeft baat bij een eerste semi-baan plus afstuderen; het bedrijf heeft baat bij een project van semi-wetenschappelijk karakter waar het anders misschien niet aan toegekomen zou zijn (een stagiaire is wel een goedkope arbeidskracht, maar dit aspect mag geen rol spelen); de begeleiders van de THE tenslotte hebben er baat bij voeling te houden met de praktijk, en het onderzoek van intelligente afstudeerders mondt menigmaal uit in publikaties, eventueel gemeenschappelijk met de begeleiders van de THE of het bedrijf. In paragraaf 3 geef ik een concreet, doorsnee voorbeeld van een afstudeerproject.

## 2. Overzicht

Deze paragraaf bestaat grotendeels uit onderstaande tabel van colleges in de OR aan de THE. Het is een dorre opsomming, maar dient als referentie. Docenten worden geacht onderzoek te doen in de onderwerpen waarover zij college geven. Lezers die meer willen weten van bepaalde docenten of onderwerpen, kunnen gemakkelijk telefoneren (040-479111).

| coll.nr.                           | docent                  | college                                                                               |
|------------------------------------|-------------------------|---------------------------------------------------------------------------------------|
| <i>Afdeling der Bedrijfskunde:</i> |                         |                                                                                       |
| 1.020.0                            | Kerbosch                | Kwantitatieve beslissingsmodellen met behulp van rekenautomaten                       |
| 1.022.6                            | Tilanus/Brocken/Janssen | Bedrijfsspel rubberschoeisel                                                          |
| 1.212.0                            | Monhemius               | Methoden van toegepast bedrijfskundig onderzoek                                       |
| 1.315.1                            | Botter/Langendam        | Projektbeheer                                                                         |
| 1.510.0                            | Wijngaard/Praagman      | Modelbouw                                                                             |
| 1.511.0                            | Tilanus/Geurts          | Operationele Research I                                                               |
| 1.512.0                            | Geraerds                | Operationele Research II                                                              |
| 1.513.1                            | Van Winkel              | Voorspellingsmethoden                                                                 |
| 1.514.3                            | Tilanus/Kroep           | Mathematische programmering                                                           |
| 1.514.6                            | Tilanus/Kroep           | Netwerkplanning                                                                       |
| 1.515.1                            | Monhemius               | Voorraad- en produktiebeheersing                                                      |
| 1.516.1                            | Bertrand                | Toepassingen van de regeltechniek in de bedrijfskunde                                 |
| 1.517.1                            | Tilanus/Puylaert        | Kwantitatieve economische methoden                                                    |
| 1.518.1                            | Geraerds/Geurts         | Onderhoudsbeheersing                                                                  |
| 1.521.1                            | Kerbosch                | Ontwerp en beoordeling van toepassings-programmatuur in de bedrijfskunde              |
| 1.526.1                            | Wijngaard               | Personeelsplanning                                                                    |
| <i>Onderafdeling der Wiskunde:</i> |                         |                                                                                       |
| 2.610.0                            | Benders                 | Optimaliseringsmethoden I                                                             |
| 2.615.0                            | Benders                 | Optimaliseringsmethoden II                                                            |
| 2.630.0                            | Benders                 | Capita optimaliseringsmethoden                                                        |
| 2.640.0                            | Benders                 | Inleiding in de mathematische programmering                                           |
| 2.650.0                            | Hautus                  | Optimalisering van regelsystemen                                                      |
| 2.651.0                            | Hautus                  | Lineaire multivariabele systemen                                                      |
| 2.660.0                            | Wessels                 | Besliskunde                                                                           |
| 2.670.0                            | Wessels                 | Stochastische beslissingsproblemen                                                    |
| 2.680.0                            | Wessels                 | Inleiding in de theorie van de stochastische processen                                |
| 2.690.0                            | Wessels                 | Stochastische processen I                                                             |
| 2.695.0                            | Van Schuppen            | Stochastische processen II                                                            |
| 2.696.0                            | Wessels                 | Wachtrij- en doorstroomanalyse in het bijzonder bij computer- en communicatiesystemen |
| 2.750.0                            | Doornbos                | Multivariate analyse                                                                  |
| 2.765.0                            | Doornbos                | Lineaire modellen                                                                     |



### 3. Van rekenen naar tellen: voorbeeld van een afstudeerproject

Een bedrijf bouwt eindprodukten op maat. Er zijn een tiental series, een 200-tal hoofdtypen, en daarbinnen nog een 40-tal opties. Van de 15000 eindprodukten die per jaar geproduceerd worden zijn er gemiddeld geen 2 dezelfde. Dat levert problemen op met de bevoorrading van ingekochte en zelf gefabriceerde onderdelen ("monodelen"). Er wordt weliswaar elke periode van 4 weken voortschrijdend een jaar vooruit geprognosticeerd. En drie perioden vooruit wordt er een vast plan gemaakt waar niet meer van mag worden afgeweken. Als we gemakshalve over maanden spreken, wordt voor bijvoorbeeld september 1981, van september 1980 tot mei 1981 iedere maand een prognose gemaakt, maar in juni 1981 moet een produktieplan vastgesteld worden. Bij deze overgang van "prognose" naar "vast" doen zich vaak grote verschuivingen voor.

Aan een student in de Bedrijfskunde werd de volgende afstudeeropdracht gegeven:

"Voor het vaststellen van de grootte van de bestellingen door de bevoorradder, wordt uitgegaan van de monodelenplanning. Deze is ingedeeld in dertien perioden waarvan de eerste drie vast zijn en de rest de prognose vormt.

Het blijkt dat bij de overgang van prognose naar vast de benodigde hoeveelheden sterk variëren. Hierdoor wordt het moeilijk om de voorraad op een optimaal niveau te houden. Het doel van het onderzoek is het bereiken van een grotere overeenkomst tussen prognose en vast, zodat hieruit een betrouwbare methodiek kan worden opgebouwd ter ondersteuning van de beslissing van de bevoorradder."

De afstudeerder kreeg een bureau midden tussen bevoorraders, en kon door veel praten en gegevens verzamelen minstens 6 belangrijke oorzaken voor de problemen van de bevoorraders vaststellen, waarbij de overgang van "prognose" naar "vast" niet de belangrijkste was:

1. Veranderingen in de prognose van maand tot maand, met name bij de overgang van prognose naar vast.
2. Veranderingen in het vaste gedeelte van de planning (wat dus officieel uitgesloten was).
3. Leveringen van monodelen buiten de planning om. De planning geldt alleen voor de fabricage van nieuwe eindprodukten en gedeeltelijk voor de afdeling Service. Daarnaast wordt ongepland geleverd aan diverse andere afdelingen.

4. Leveranciers van ingekochte monodelen overschrijden de levertijden.

5. Er wordt met twee soorten stuklijsten gewerkt: een constructie-stuklijst en een produktielijst. De eerste is op de verkoop gericht en de laatste is op de produktie gericht. Het is onmogelijk te voorkomen dat er verschillen optreden tussen de beide stuklijsten. (In de loop van het onderzoeksproject is afgestapt van het werken met twee stuklijsten.)

6. Telfershillen. Dit is een eufemisme voor foutieve registratie in het computersysteem van de werkelijke voorraden. De zogenaamde telfershillen zijn vaak cumulatief. Als er bijvoorbeeld volgens de computer één soort monodelen gebruikt wordt en in werkelijkheid een ander, dan loopt de voorraad van het ene monodeel tot ongekende hoogte op, terwijl over het andere monodeel de bevoorraders plotseling een "nulmelding" ontvangen.

De afstudeerder heeft een en ander gekwantificeerd en doorgerekend. Het bleek dat oorzaak nr. 6 verreweg het belangrijkste was. Hij heeft organisatorische veranderingen doorgesproken, een nieuwe categorie-indeling van de monodelen gemaakt (een veredelde ABC-analyse), en optimale veiligheidsvoorraden berekend. Zijn belangrijkste aanbeveling was om het aantal tellers uit te breiden van 4 tot 10 personen. (Het aantal 10 was berekend als zijnde optimaal.) Die tellers moeten niets anders doen dan de "telfershillen" tussen werkelijke en geregistreerde voorraad uit de weg ruimen. In totaal zou 1 miljoen gulden per jaar op voorraadkosten kunnen worden bespaard [1].

Twee aspecten vind ik interessant aan dit afstudeerproject. Ten eerste hoezeer in de loop van een onderzoek het zwaartepunt van de belangstelling kan verschuiven, ook van kwantitatief naar kwalitatief, en omgekeerd. Ten tweede dat het resultaat van veel statistische gegevensverwerking en berekeningen de simpele aanbeveling kan zijn: meer tellen!

#### Literatuur

- [1] H.N. Post, "Voorraadbeheersing voor de ...", Afstudeerrapport, Afdeling Bedrijfskunde, Technische Hogeschool Eindhoven, 1980.