Het LLL-algoritme

een case-study naar wiskunde
en technologische innovatie

Frank Buijnsters
Tim van Lent
Wouter van Orsouw

Project Modellenpracticum 2010
Begeleider: Prof. dr. Nicolaas Landsman

Radboud Universiteit Nijmegen
23 juli 2010

Inhoudsopgave

1 Inleiding

Wat maakt wetenschap tot wetenschap? Waarin verschilt het van andere, meer alledaagse vormen van menselijk redeneren? Het is een vraag waarover talloze filosofen, wetenschappers en andere intellectuele types zich het hoofd hebben gebroken, met evenzoveel verschillende antwoorden tot gevolg. Van de vele gunstige kenmerken die in de geschiedenis aan de onderneming zijn toegeschreven, zoals objectiviteit, empirisme, precisie en waarheidsdrang, springt er een het meest in het oog: diepgang. In tegenstelling tot politici of managers bijten wetenschappers zich vast in zeer nauw omschreven vragen. Ze zijn pas tevreden als ze het probleem werkelijk doorgronden. De wetenschap is heel lang bezig met heel weinig.

Dat is meteen haar grootste probleem. Welke kwesties zijn zo veel hoogintelligente aandacht waard? De wereld is vreselijk groot en complex. Selectiviteit is dus een noodzaak. Waarop is die selectie gebaseerd? Deze vraag is zo wezenlijk dat wetenschappelijk onderzoek vaak wordt ingedeeld naar de manier waarop zij haar beantwoordt. Zowel het lineaire model als het kwadrantenmodel zijn daarvan een voorbeeld (zie hoofdstuk 2). Het lineaire model maakt een (gradueel) onderscheid tussen *toegepaste wetenschap*, die problemen probeert op te lossen van een groot en direct economisch, politiek of anderszins maatschappelijk belang, en *zuivere wetenschap*, die streeft naar de ontwikkeling en verfijning van algemeen toepasselijke theorieën en modellen ('begrip') waarvan de waarde wordt gevonden in de interne logica van het vakgebied. In de zuivere natuurkunde, of de zuivere natuurwetenschap in het algemeen, vindt deze taxatie voornamelijk plaats in termen van de verklarende en voorspellende kracht van natuurlijke fenomenen en het aantal aannames dat nodig is om deze kracht te bereiken. De wiskunde, daarentegen, bestudeert abstracties en axiomatische systemen waarvan vele geen directe fysische pendant hebben, of lijken te hebben. Er is *a priori* vrijwel geen beperking aan de onderwerpen die interessant zouden kunnen zijn. Dat maakt de vraag van selectie voor de wiskunde nog fundamenteler.

Het is een vraag met politieke implicaties. De staat bekostigt immers een aanzienlijk deel van het wetenschappelijk onderzoek. Na de Tweede Wereldoorlog, waarin de Verenigde Staten en andere landen, met het oog op militaire toepassingen, voor het eerst op zeer grote schaal onderzoeksprojecten ontplooiden, hebben veel landen blijvende stelsels voor onderzoeksfinanciering opgezet. Deze stelsels hadden uitdrukkelijk als doel

niet alleen onderzoek van direct militair nut te bekostigen, maar vooral ook de zuivere wetenschap. Een van de meest uitgesproken voorstanders van deze opzet was Vannevar Bush, die, ondersteund met wat later het lineaire model is genoemd, de stelling verdedigde dat de Verenigde Staten hun technologische koppositie alleen konden behouden door geld te investeren in zuiver wetenschappelijk onderzoek dat op geen enkele manier wordt beïnvloed door toepassingsgerichte overwegingen[8]. Het is precies dit standpunt dat Donald E. Stokes aanvalt in zijn boek *Pasteur's Quadrant* [2]. Kern van zijn betoog is dat de dichotomie die Bush ziet tussen toegepaste wetenschap ('het kwadrant van Edison') en zuivere wetenschap ('het kwadrant van Bohr') vals is: er zijn ook talloze voorbeelden van wetenschappers en wetenschappelijk onderzoek dat gemotiveerd is door zowel een drang naar wezenlijk nieuw begrip als naar praktisch bruikbare toepassingen. Stokes noemt dit het kwadrant van Pasteur.

Stokes gaat in zijn boek vooral in op voorbeelden uit de natuurwetenschap en de techniek. Wij hebben geprobeerd te achterhalen hoe toepasbaar het lineaire model en het kwadrantenmodel zijn op de wiskunde. Daarbij hebben we, bij wijze van *case study*, de ontwikkeling van het LLL-algoritme als uitgangspunt genomen. Hoe past deze casus binnen de beide modellen?

Ten behoeve van dit onderzoek hebben we twee interviews afgenomen. Het eerste was met Hendrik Lenstra, hoogleraar in Leiden en één van de drie 'L's', het tweede met Ionica Smeets, onderzoeker, wetenschapsjournalist en Wiskundemeisje[1]. Zij heeft zich beziggehouden met het LLL-algoritme, allereerst omdat ze het nodig had voor haar promotieonderzoek. Daarnaast heeft ze een artikel geschreven over de ontwikkeling van het algoritme. Dit artikel is helaas niet, zoals aanvankelijk beoogd, gepubliceerd in een populairwetenschappelijk tijdschrift, maar wel in het boek *The LLL Algorithm: Survey and Applications*[1], dat is uitgegeven ter gelegenheid van het vijfentwintigjarig jubileum van het LLL-algoritme. Korte uitwerkingen van de interviews zijn als appendix toegevoegd.

[1] www.wiskundemeisjes.nl

2 Modellen voor wetenschap en innovatie

De subsidiëring van wetenschappelijk onderzoek, als alle overheidsbestedingen, is onderwerp van voortdurende politieke discussie. Steeds vaker en steeds opener vragen beleidsmakers en het publiek zich af of alle investeringen in de wetenschap het land ook wat opleveren. De vertrouwensband tussen de wetenschap en de overheid is uit het lood geslagen.

Dat is althans de premisse die de aanleiding vormt van het boek *Pasteur's Quadrant: Basic Science and Technological Innovation* (1997) van de gerespecteerde Amerikaanse politicoloog en beleidsmaker Donald E. Stokes (1927–1997). Zijn boek gaat uitgebreid in op zowel de geschiedenis van onderzoeksfinanciering als het huidige beleid, met name in de Verenigde Staten, en betoogt dat de problemen samenhangen met onderliggende gebreken in de visie van velen, uit zowel de politieke als de wetenschappelijke wereld, op de relatie tussen wetenschappelijke vooruitgang en technologische innovatie. Deze visie, aangeduid als het 'lineaire model', vindt zijn oorsprong in de opvattingen van een invloedrijke Amerikaanse ingenieur en bestuurder, Vannevar Bush, en met name in zijn notitie *Science, the Endless Frontier* (1945).

Bush schreef dit betoog, gepubliceerd ongeveer twee weken voor ontploffing van de atoombommen op Hiroshima en Nagasaki, in opdracht van president Franklin D. Roosevelt, die wilde weten hoe de vaart die de Tweede Wereldoorlog aan wetenschappelijke en technische ontwikkeling had gegeven kon worden behouden en benut in naoorlogs Amerika. Bush is een vurig pleitbezorger van structurele financiering van wetenschappelijk onderzoek door de staat. Waarom zouden marktpartijen deze financiering niet op zich kunnen nemen? Hoewel ook R&D-laboratoria, marktonderzoeksbureaus en verzekeraars onderzoek verrichten, zijn hun investeringen vrijwel altijd sterk toepassingsgericht: alleen kennis die patenteerbaar is, specifiek nuttig is voor het eigen bedrijf of expertise oplevert die sneller is in te zetten dan andere partijen haar kunnen overnemen brengt een concurrentievoordeel met zich mee. Het gevolg, zo stelt Bush: "Applied science invariably drives out pure." Toch is het juist de zuivere wetenschap, "performed without thought of practical ends," die uiteindelijk aan de basis staat van de meest wezenlijke innovatie. Aan de overheid dus de schone taak om het zuiver wetenschappelijke onderzoek te subsidiëren – liefst zonder zich al te veel met de inhoud te bemoeien.

In zijn notitie doet Bush gedetailleerde voorstellen voor een 'National Research Foundation'. Ze zijn nooit in die vorm uitgevoerd, maar zijn ideeën hebben wel aan de basis gestaan van de grootschalige organsiaties, zoals de National Science Foundation in de Verenigde Staten, die overheden van veel geïndustrialiseerde landen na de Tweede Wereldoorlog hebben opgezet.

Statisch en dynamisch

In *Pasteur's Quadrant* noemt Stokes de prestatie van Bush bewonderenswaardig. Niet alleen was zijn timing perfect – de afloop van de oorlog maakte de bevolking als nooit tevoren bewust van het enorme belang van wetenschap en techniek voor het succes van de natie – hij baseerde zijn betoog, zo beargumenteert Stokes uitvoerig in hoofdstuk 2 van zijn boek, op ideeën die een sterke weerklank vinden in de Westerse geschiedenis en filosofie. Het resultaat was ernaar. De opvattingen van Bush ontwikkelden zich tot het 'lineare model', een manier van denken over wetenschap en innovatie die veel, misschien wel alle, wetenschappers en makers van wetenschapsbeleid nu bewust of onbewust met zich meedragen.

Stokes onderscheidt twee gerelateerde maar verschillende 'verschijningsvormen' van het lineaire model. In zijn 'statische vorm', gebaseerd op Bush' bewering dat zuivere wetenschap wordt bedreven zonder afleidende gedachten over mogelijk praktisch nut, stelt het model dat er een eendimensionale schaal bestaat die zuivere wetenschap van toegepaste wetenschap onderscheidt. Aan de ene kant bevinden zich de wetenschappers die, als in het klassiek Griekse ideaal van 'kennis om de kennis', op zoek gaan naar fundamenteel inzicht en wezenlijk nieuwe theorieën, zonder zich zelfs maar af te vragen of deze kennis ooit een maatschappelijk nut zal hebben. Aan de andere kant van het spectrum staan onderzoekers die, als de aquaductbouwers of de loodgieters van het oude Rome, streven naar resultaten die direct leiden tot economisch of maatschappelijk nuttige toepassingen. Natuurlijk zijn er ook onderzoekers die zich ergens tussen beide extremen in bevinden, maar – en dit is het wezenlijke punt – een meer toepassingsgerichte wetenschapper is automatisch minder geïnteresseerd in fundamenteel nieuwe kennis, en vice versa.

De 'dynamische vorm' van het lineaire model is gebaseerd op Bush' bewering dat "basic research is the pacemaker of technological progress" en dat "a nation which depends upon others for its new basic scientific knowledge will be slow in its industrial progress and weak in its competitive

postition in world trade." In het lineaire model komt alle innovatie van betekenis voort uit nieuwe zuiver wetenschappelijke kennis. Het model wordt vaak weergegeven als een stoomschema dat begint bij 'zuiver wetenschappelijk onderzoek' en, langs een keten van pijlen, via 'toegepast wetenschappelijk onderzoek' en 'ontwikkeling' uiteindelijk uitkomt bij 'productie en uitvoering'. Er is geen kruisbestuiving tussen de verschillende stappen: de kennisstroom gaat *van* de wetenschap *naar* toepassingen.

Moeilijk verkoopbaar

Wat is volgens Stokes zo problematisch aan deze visie? Eerst en vooral: zij simpelweg onjuist. Stokes geeft tal van voorbeelden van wetenschappers met een sterk zuiver wetenschappelijke interesse en motivatie die zich tegelijkertijd intensief bezighielden met problemen uit de industriële, medische of economische praktijk. Sterker: een praktisch probleem was voor hun belangrijkste fundamentele ontdekkingen vaak de aanleiding. Het toonbeeld van dit soort onderzoek levert Louis Pasteur, de Fransman die grote bekendheid verwierf met zijn ontdekking van het 'pasteuriseren', 'steriliseren' en zijn vaccin tegen hondsdolheid. Deze heel nuttige ontekkingen kwamen echter voort uit een fundamenteel inzicht: infectieziekte, plaag en bederf onstaan niet vanzelf maar worden veroorzaakt door microben. Zijn experimenten brachten de genadeklap toe aan de theorie van *generatio spontanea*, het idee dat schimmels, maden en andere 'lagere' levensvormen vanzelf ontstaan uit geschikte dode materie. Als zuivere wetenschapper plaatsen deze inzichten Pasteur op dezelfde hoogte als Antonie van Leeuwenhoek, Gregor Mendel of James D. Watson en Francis Crick. Waar moeten we zó een onderzoeker plaatsen op die eendimensionale schaal uit de statische vorm van het lineaire model? Ergens in het midden van de lijn, zou je kunnen zeggen, maar Stokes vindt dat dit geen recht doet aan Pasteurs werk, dat niet 'allebei een beetje' was maar 'allebei heel veel'.

Ook het dynamische aspect van het lineaire model verdient voor Stokes een veel kritischere beschouwing dan het doorgaans krijgt. Natuurlijk, het is een enorme versimpeling – dat hebben anderen vóór Stokes al opgemerkt – maar dat is niet eens het grootste manco. Ten eerste lijkt het niet juist dat alle technologische innovatie voortkomt uit wetenschap (het 'pacemaker'-idee van Bush). Ook al is de meeste techniek pas goed te begrijpen met wetenschap, vaak ging de uitvinding *vooraf* aan de fundamentele kennis die haar werking verklaart. Bovendien hebben landen als Japan hun sterke

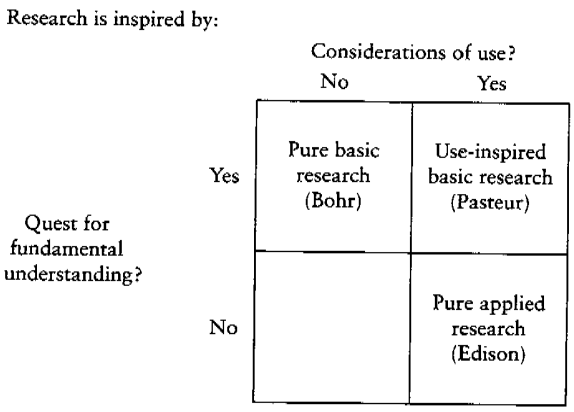
marktpositie in de autoindustrie en de consumentenelektronica meer te danken aan een voortdurende stroom snelle, kleine verbeteringen in hun producten dan aan nieuwe fundamentele kennis.

Ten tweede – en dit is nog wezenlijker – is veel zuiver wetenschappelijk onderzoek alleen mogelijk dankzij de nieuwe techniek; sommige zuivere wetenschapsgebieden ontlenen daaraan zelfs hun bestaansrecht. Stokes noemt de vastestoffysica als voorbeeld: deze wetenschap werd en wordt voor een aanmerkelijk deel gevoed door de wens beter te begrijpen hoe halfgeleiders werken en te verbeteren zijn. Opnieuw kan ook Pasteur als voorbeeld fungeren. Brouwers of wijnmakers vroegen Pasteur soms hoe zij de hardnekkige problemen konden oplossen die zich voordeden in het vergistingsproces. Het lijkt erop dat Pasteur ze niet alleen kon *helpen* met zijn theorie over micro-organismen, maar dat deze praktische problemen voor zijn theorie ook een belangrijke inspiratiebron waren. Pasteur moedigde ook zijn studenten aan voortdurend in contact te blijven met de praktijk.

Er is nog een heel andere reden voor kritiek op het lineaire model. Als wetenschapsfilosofische theorie is het model misschien te simpel, maar dat zou zo'n probleem niet zijn als het kon bijdragen aan een brede acceptatie van het nut van wetenschap bij het publiek. Maar volgens Stokes is het tegendeel waar. Na de Tweede Wereldoorlog voelden de Verenigde Staten zich nog de hegemoniale natie en de grootste technologische macht. In zo'n klimaat is het aannemelijk dat de langetermijninvesteringen in wetenschap ook voornamelijk tot voordeel van het eigen land zullen zijn. De opkomst van Japan en, meer recentelijk, China en India maken dit verhaal moeilijker verkoopbaar: stimuleren onze mooie wetenschappelijke resultaten dadelijk niet de Chinese innovatiekracht? Daar komt nog bij dat kiezers in het algemeen tegenwoordig misschien een sterkere anti-establishmenthouding hebben dan vroeger.

Visie

Wat is het alternatief? In hoofdstuk 3 van zijn boek doet Stokes zijn eigen visie op het verband tussen zuivere wetenschap en innovatie uit de doeken. In plaats van één dimensie breidt Stokes zijn model uit tot twee (zie figuur 1.) Stokes' model is opgedeeld in vier kwadranten. Twee daarvan, 'het kwadrant van Bohr' en 'het kwadrant van Edison', zijn vergelijkbaar met de categorieën 'zuiver wetenschappelijk onderzoek' en 'toegepaste wetenschap' uit het lineaire model. Stokes voegt daar 'het kwadrant van Pasteur'



Figuur 1: Het kwadrantenmodel van Stokes. Overgenomen uit [2].

aan toe: onderzoek dat gemotiveerd is door zowel een hang naar nieuw fundamenteel inzicht als de hoop op praktische toepassingen. Eén van de vier kwadranten blijft in Stokes' model opvallend leeg: het onwaarschijnlijk klinkende onderzoek dat noch een fundamenteel, noch een toegepast doel heeft. Haalt dit het kwadrantenmodel niet een beetje onderuit? Stokes probeert deze kritiek te pareren door te stellen dat zulk onderzoek wel degelijk bestaat. Te denken valt aan het vergaren van 'kennis om de kennis' waaraan geen fundamenteel belang voor de theorie kan worden toegeschreven, zoals het samenstellen van de Flora en andere activiteiten die natuurkundigen, in navolging van Ernest Rutherford, graag mogen afdoen als 'postzegels verzamelen'.

In de laatste twee hoofdstukken van zijn boek gaat Stokes uitvoerig in op de vraag wat zijn kwadrantenmodel zou kunnen bijdragen aan een herstel van vertrouwen tussen wetenschappers, beleidsmakers en het publiek. Ook geeft hij een gedetailleerde uiteenzetting van de recente en zeer recente geschiedenis van het wetenschapsbeleid in de Verenigde Staten. In zijn visie is er zeker ruimte voor werk in het kwadrant van Bohr. Maar door daarnaast meer aandacht te geven aan het kwadrant van Pasteur, of het bestaan van dat soort onderzoek op zijn minst te erkennen en beter voor het voetlicht te brengen, hoopt hij dat het publiek en de politiek sterker overtuigd raken van het belang van wetenschap in het algemeen – dus ook van wetenschap met een fundamenteel doel. Hem lijkt dit effectiever dan blijven vasthouden aan

het bekende verhaal dat zuivere wetenschap 'ooit ergens goed voor zal zijn, al weten we nog niet wat.' Tegelijkertijd hoopt hij de wetenschappelijke wereld, die haar academische vrijheid altijd met verve heeft verdedigd, ervan te overtuigen dat het sterker benadrukken van een toepassingsgericht doel niet noodzakelijkerwijs minder aandacht betekent voor het opdoen van fundamentele kennis. Integendeel: het erkennen van de wisselwerking tussen theorie en toepassingen kan ertoe leiden dat in toepassingsgerichte projecten als de kernfusiereactor óók voldoende investeringen worden gedaan in het verstevigen van het noodzakelijke theoretisch fundament.

3 Het LLL-algoritme

In dit hoofdstuk bekijken we de wiskunde achter het LLL-algoritme. Het uiteindelijke algoritme is gepubliceerd in het artikel *Factoring Polynomials with Rational coefficients* uit 1982[4]. Hieraan voorafgaand was er enige communicatie geweest tussen Hendrik Lenstra en Lovász. Lenstra had namelijk al een algoritme ontwikkeld om een basis van een rooster te reduceren. Lovász had dit echter verbeterd en dat is bekend geworden als het LLL-algoritme. Het algoritme vindt een gereduceerde basis in een rooster. Om in te zien hoe het werkt moeten we eerst weten wat 'gereduceerd' precies inhoudt.

3.1 LLL-gereduceerd

Het LLL-algoritme is een basisreductiealgoritme voor roosters.

Definitie: Een deelverzameling L van de n-dimensionale vectorruimte $\mathbb{R}^n$ heet een *rooster* als er een basis $b_1, b_2, ...b_n$ van $\mathbb{R}^n$ bestaat zo dat

$$L = \sum_{i=1}^{n} \mathbb{Z} b_i = \{\sum_{i=1}^{n} r_i b_i | r_i \in \mathbb{Z} \text{ voor } 1 \leq i \leq n\}$$

In dit geval noemen we $b = (b_1, b_2, ...b_n)$ een *basis* voor L.

Met behulp van het Gram-Schmidtproces kunnen van een basis b een orthogonale basis b^* maken. Dit proces is op de volgende manier inductief gedefinieerd:

$$\begin{aligned} b_i^* &= b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j \\ \mu_{ij} &= \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle} \end{aligned}$$

Definitie: Een basis $b = (b_1, b_2, ...b_n)$ voor een rooster L heet *LLL-gereduceerd* als

$$|\mu_{ij}| \leq \frac{1}{2} \text{ voor } 1 \leq j < i \leq n \tag{1}$$

en

$$|b_i^* + \mu_{ii-1} b_{i-1}^*|^2 \geq \frac{3}{4}|b_{i-1}^*|^2 \text{ voor } 1 < i \leq n \tag{2}$$

Het LLL-algoritme vindt voor een gegeven basis b een gereduceerde basis b'. De constante $\frac{3}{4}$ is willekeurig en kan vervangen worden door een vaste $y \in \mathbb{R}$ met $\frac{1}{4} < y < 1$.
Uit de eisen (1) en (2) is in te zien dat in een gereduceerde basis de basisvectoren niet te veel in lengte verschillen en dat de eerste basisvector relatief klein is. Dit volgt uit de volgende propositie.

Propositie 3.1. *Zij $b_1, b_2, ...b_n$ een gereduceerde basis in een rooster L in $\mathbb{R}$ met $b_1*, b_2^*, ...b_n^*$ gedefinieerd zoals hierboven. Dan geldt dat*

$$|b_j|^2 \leq 2^{i-1}|b_i^*|^2 \text{ voor } 1 \leq j \leq i \leq n \tag{3}$$

en

$$|b_1| \leq 2^{\frac{n-1}{4}} \cdot d(L)^{\frac{1}{n}} \tag{4}$$

waarbij $d(L)$ de determinant van het rooster is.

Bewijs: Uit de definitie van 'gereduceerd' volgt dat

$$|b_j^*|^2 \geq (\frac{3}{4} - \mu_{ii-1}^2)|b_{i-1}|^2 \geq \frac{1}{2}|b_{i-1}|^2$$

voor $1 < i \leq n$. Dus passen we inductie toe, dan volgt dat

$$|b_j^*|^2 \leq 2^{i-j}|b_i^*|^2 \text{ voor } 1 \leq j \leq i \leq n.$$

Nu gebruiken we wederom de definitie van 'gereduceerd' en tevens de definitie van b_i^*. Dan zien we in dat

$$\begin{aligned} |b_i|^2 &= |b_i^*|^2 + \sum_{j=1}^{i-1} \mu_{ij}^2 |b_j^*|^2 \\ &\leq |b_i^*|^2 + \sum_{j=1}^{i-1} \frac{1}{4} 2^{i-j} |b_i^*|^2 \\ &= (1 + \frac{1}{4}(2^i - 2))|b_i^*|^2 \\ &\leq 2^{i-1}|b_i^*|^2 \end{aligned}$$

Maar dan volgt

$$|b_j|^2 \leq 2^{j-1}|b_j^*|^2 \leq 2^{i-1}|b_i^*|^2$$

Verder hebben we dat $d(L) = |\det(b_1^*, b_2^*, ...b_n^*)|$. Omdat de b_i^*'s een orthogonale basis vormen volgt dat

$$d(L) = \prod_{i=1}^{n} |b_i^*|.$$

Uit voorgaande berekening volgt dan eenvoudig dat $|b_1| \leq 2^{\frac{n-1}{4}} \cdot d(L)^{\frac{1}{n}}$. $\square$

Verder kunnen we laten zien dat de andere gereduceerde basisvectoren niet te veel afwijken van deze eerste basisvector.

3.2 Het algoritme

Het LLL-algoritme werkt als volgt: Bereken μ_{ij} en b_i^*, zoals hierboven gedefinieerd. Begin met $k = 2$.[2]

1. Reduceer de lengte van vector b_k door er $\lfloor 0,5 + \mu_{kk-1} \rfloor$ keer de vector b_{k-1} er vanaf te halen.

2. Kijk nu of b_k en b_{k-1} voldoen aan de eisen van een gereduceerde basis.
   Zo ja, ga naar 3.
   Zo nee, ga naar 4.

3. Reduceer b_k door er $\lfloor 0,5 + \mu_{kl} \rfloor$ keer de vector b_l er vanaf te halen, voor $l = k-2, ...2, 1$.
   Geldt nu $k = n$: Zo ja, Stop. Zo nee, begin opnieuw met $k \to k+1$.

4. Verwissel de vectoren b_k en b_{k-1} en ga door met $k \to k-1$.

Hierbij is het van belang dat de vectoren b_i en de getallen μ_{ij} constant worden bijgewerkt wanneer de vectoren veranderen.

[2] In stadium k voldoen de vectoren $b_1, ...b_{k-1}$ aan de eerste eis van een gereduceerde basis.

4 Geschiedenis van het algoritme

Het LLL-algoritme is begonnen met een vraag aan Hendrik Lenstra gesteld door Peter van Emde Boas. Van Emde Boas was in die tijd (rond 1980) bezig met een probleem samen met Alberto Marchetti-Spaccamela. Hij had de vraag: "Gegeven drie punten met rationale coördinaten in een vlak, is het mogelijk om in polynomiale tijd te bepalen of er een punt bestaat met gehele coëfficiënten dat binnen de driehoek van deze drie punten ligt?" Hendrik Lenstra wist dat dit probleem allang opgelost was door Gauss. Het antwoord is 'ja'.

Hendrik Lenstra dacht dat het hiermee opgelost was, maar van Emde Boas kwam enkele maanden later weer bij Hendrik Lenstra met een vergelijkbare vraag. Dit keer kwam hij samen Marchetti. Lenstra dacht in eerste instantie dat hij dit al had opgelost. Maar de vraag was dit keer of het op eenzelfde manier ook werkte in meerdere dimensies. Na wat er na hebben gekeken kwam Hendrik Lenstra, wederom, vrij snel met het antwoord 'ja'. Hierdoor kwam Lenstra's algoritme voor geheeltallig lineair programmeren tot stand. Hieronder nog even voor de duidelijkheid de formulering van de tweede vraag:

Vraag 2. *Stel n en m zijn positieve reële getallen, zij A een $m \times n$ matrix met gehele getallen als invoer en $b \in \mathbb{Z}^m$.*
Is er een vector $x \in \mathbb{Z}^n$ met $Ax \leq b$, zo dat $\mathbb{Z}^n \cap K$ leeg is als $K = \{x \in \mathbb{R}^n | Ax \leq b\}$?

Lineair programmeren is ook wel bekend als lineaire optimalisatie. Dat is het probleem van maximalisering of minimalisering van een lineaire functie over convexe veelvlakken, die voortgebracht zijn door niet-negatieve lineaire randvoorwaarden. Hendrik Lenstra was hier eigenlijk helemaal niet mee bezig. Zo zei hij: "Zonder de vraag over de driehoeken in geheeltallige roosters was ik er nooit opgekomen."

Toen László Lovász zich ermee ging bezighouden, was hij met iets anders bezig. Zijn intentie was helemaal niet het algoritme te verbeteren (wat hij uiteindelijk wel deed), maar hij was bezig met een detail in de ellipsoïdemethode. Lovász zei hierover: "De ellipsoïdemethode is ontwikkeld door Sovjetonderzoekers eind jaren 70. Khachiyan merkte op dat het algoritme kon worden gebruikt om lineaire programmeerproblemen op te lossen in polynomiale tijd, wat een groot onopgelost probleem was."

Lovász leefde in Hongarije, maar kon slechts af en toe reizen. Het IJzeren Gordijn gordijn maakte dat moeilijk. Eind jaren zeventig was hij

in Canada en in de zomer in Stanford. Daar ontmoette hij Peter Gács en daar had iemand het artikel van Khachiyan naar hen verzonden. Toen hij weer terugging naar Hongarije zag hij nog de mogelijkheid om Amsterdam en Bonn te bezoeken. In Amsterdam ontmoette hij Lex Schrijver en in Bonn Martin Grötschel. Hij besprak met hen de ellipsoïdemethode, waarover zij enthousiast werden. Ze hebben er uiteindelijk een boek over geschreven, maar kwamen erachter dat één ding ontbrak. In Bonn had Lovász toevalligerwijs een lezing van Hendrik Lenstra over geheeltallig programmeren gehoord. Hierin kwam het probleem van het zoeken naar korte vectoren in roosters naar voren. Lovász bedacht dat dit precies de stap was die ze misten.

Hebberig

Ongeveer gelijktijdig kwam ook Arjen Lenstra, broer van Hendrik en nog student, met het basisreductieprobleem in aanraking. Hij was bezig met zijn scriptie over polynoomfactorisatie over algebraïsche getallenlichamen. Dit werd over het algemeen gedaan met de Berlekamp-Henselmethode. Maar er was een groot probleem, want in de laatste stap kon de rekentijd exponentieel worden in de graad van het polynoom. Iedereen had zichzelf ervan overtuigd dat het probleem waarschijnlijk echt niet in polynomiale tijd oplosbaar was: men was juist bezig met een zoektocht naar polynomen waar de Belekamp-Henselmethode ontzettend lang over deed.

Deel van de opdracht van Arjens scriptie, waarvan Peter van Emde Boas de begeleider was, was deze methode te implementeren. Toen Arjen Lenstra het hierover had met zijn broer Hendrik Lenstra, zag Hendrik een stap die je beter kon vervangen door een roosterreductiestap. Arjen en Hendrik kwamen erachter dat de cruciale, niet-polynomiale stap in Berlekamp-Hensel vervangen kon worden door kortstevectorberekeningen. En deze konden weer vervangen worden door roosterreductie. Leuk maar nutteloos, dacht men. Want het vinden van de kortste vector noch de basisreductie verliep in polynomiale tijd. Arjen had wel geobserveerd dat het algoritme sneller kon worden gemaakt. Maar niemand leek er iets om te geven, want het was toch exponentieel over de graad van het polynoom. Arjen: “Ik gebruikte de roosterreductie van Hendriks artikel over geheeltallig lineair programmeren. Dit reductiealgoritme werkte niet in polynomiale tijd, maar wat maakt dat uit? Het oorspronkelijke probleem was al exponentieel, dus wat verlies ik ermee? Dus werd het algoritme geïmplementeerd, en het werkte prachtig.”

Van dit alles was Lovász niet op de hoogte toen hij nog wat zat na te

denken over de lezing van Hendrik Lenstra in Bonn. Hem leek dat er nog wel wat te verbeteren viel aan het basisreductiealgoritme dat hem juist was gepresenteerd. Snel begon hij te brainstormen, te schrijven en te prutsen met zijn rekenmachine. Lovász kwam erachter dat het beter was om niet te 'hebberig' te zijn in het basisreductiealgoritme. Waar Hendrik Lenstra gedurende het algoritme telkens de kortste vector uit de basis vooraan in de rij zette, begreep Lovász dat je beter alleen naburige vectoren kon verwisselen en dan nog enkel wanneer er, in zekere zin, duidelijk voortgang mee kan worden geboekt. Lenstra zegt hierover: "Ik had een zeer naïeve manier voor het vinden van de benodigde transformatie, waarbij deze methode alleen polynomiaal[3] is voor vaste dimensie n. Lovász vond een algoritme om dit te doen in polynomiale tijd voor variabele n."

Lovász stuurde een brief over zijn verbeterde roosterreductie in polynomiale tijd naar Henkdrik Lenstra. Die begon direct met zoeken naar fouten in zijn bewijs. Lenstra wist immers dat zo'n algoritme de niet-polynomiale stap in het polynoomfactorisatiealgoritme kon vervangen. In een brief aan Lovász zei hij: "Sinds ik jouw brief heb ontvangen ben ik verbaasd, want het blijkt dat uit jouw basisreductiealgoritme volgt dat er een polynomiaal algoritme is voor factorisatie in $\mathbb{Q}[X]$. Ik ben op zoek gegaan naar fouten in het bewijs, maar ik heb er nog geen gevonden." Men was zo overtuigd dat het factoriseren niet kon in polynomiale tijd (aangezien getallen ontbinden in priemfactoren niet polynomiaal was) dat ze dachten dat ze wel ergens een fout zouden hebben gemaakt. Die bleek er niet te zijn en zo kwamen ze plotseling tot de conclusie dat polynomen met rationale coëfficiënten konden worden ontbonden in polynomiale tijd.

[3]Deze polynomialiteit zit hem in de afhankelijkheid van het aantal operaties van de lengte in bits van de basisvectoren. De te reduceren basisvectoren worden gegeven in $\mathbb{Z}^n$ of $\mathbb{Q}^n$. Daaraan kan dus een eindig aantal bits worden toegekend. Hendrik Lenstra's algoritme was weliswaar polynomiaal in deze *lengte* van de basisvectoren, maar niet in hun *aantal*.

5 Complexiteit en $P = NP$

Bij een algoritme vraagt men zich af of het snel werkt. Het is namelijk niet gewenst dat het heel erg lang duurt voordat het algoritme klaar is met rekenen. De maat voor de snelheid van een algoritme is de complexiteit. De complexititeit van een algoritme is de hoeveelheid elementaire operaties die nodig zijn tot het algoritme termineert. Dit aantal operaties is uitgedrukt als functie van de *lengte van de invoer*. Deze lengte is het aantal bits dat nodig is om de invoer te representeren. In het algemeen wordt de invoer k binair gerepresenteerd. De lengte van de invoer is dan ${}^2\log(k) = n$.

De complexiteit van een algoritme is niet het aantal operaties, maar de orde hiervan. We gebruiken de *grote-O-notatie* voor de orde. Een functie $f(x)$ heeft orde $g(x)$, genoteerd als $f(x) = O(g(x))$, dan en slechts dan als er een constante M bestaat zo dat $f(x) \leq M \cdot g(x)$, voor alle x groter dan een zekere x_0. Een algoritme heeft een polynomiale complexiteit als de rekentijd $T(n)$ een polynoom in de lengte van de invoer als bovengrens heeft. Kort gezegd: $T(n) = O(n^m)$ voor een zekere constante m.

5.1 P en NP

Er bestaan verschillende klassen van algoritmen, afhankelijk van de complexiteit. Dit zijn de complexiteitsklassen. De twee belangrijkste complexiteitsklassen zijn P en NP. Voor we verder kunnen ingaan op deze klassen moeten we kijken naar het soort problemen. Er zijn *beslisproblemen* en *zoekproblemen*. Beslisproblemen zijn vragen waarop met 'ja' en 'nee' kan worden geantwoord. Een eenvoudig voorbeeld is: "Voor $x, y \in \mathbb{N}$, is x een deler van y?" In het geval van een zoekprobleem willen we ook echt een oplossing voor het probleem vinden. Een voorbeeld is de vraag: "Wat is x gedeeld door y?"

De klasse van zoekproblemen is equivalent aan de klasse van beslisproblemen. Voor ieder zoekprobleem kunnen we namelijk een beslisprobleem maken. Andersom kunnen we bij ieder beslisprobleem een zoekprobleem maken. Aan de bovenstaande voorbeelden is te zien hoe we van de ene klasse naar de andere kunnen overstappen. De complexiteitsklasse P is gedefinieerd als de klasse van alle beslisproblemen die in polynomiale tijd zijn op te lossen met een deterministische Turingmachine. P staat voor deterministic Polynomial time.

De complexiteitsklasse NP, daarentegen, is gedefinieerd als de klasse

van alle beslisproblemen die in polynomiale tijd zijn op te lossen met een niet-deterministische Turingmachine. Voor een probleem in deze klasse geldt dat een oplossing niet altijd in polynomiale tijd is te vinden, maar wel in polynomiale tijd te *verifiëren*. Dus er bestaat een deterministische Turingmachine die in polynomiale tijd laat zien dat een waarde x een oplossing is voor het beslisprobleem. NP staat voor Non-deterministic Polynomial time. Merk op dat P een deelklasse is van NP.

5.2 $P = NP$?

Een onopgelost probleem in de theoretische informatica is de vraag of de klassen P gelijk is aan de klasse NP. Dit is kort samen te vatten in de vraag: "Als ja-antwoorden op een ja/nee-vraag 'snel' kunnen worden geverifiëerd, kunnen de antwoorden dan ook 'snel' gevonden worden?" Met snel bedoelen we 'in polynomiale tijd'. Een polynomiaal algoritme is over het algemeen ook snel in de praktijk, maar dit is niet altijd zo. De vraag of P gelijk is aan NP is zelfs één van de Millennium Prize Problems. Het wordt dus gezien als één van de belangrijkste problemen voor de wiskunde in de 21e eeuw. Een belangrijke klasse voor deze vraag is de klasse NP-compleet. Een probleem Q is een NP-compleet probleem als het een NP probleem is, zodanig dat ieder ander probleem in NP is te reduceren tot Q. Dit reduceren moet echter wel in polynomiale tijd gebeuren.

We zien dus dat ook NP-compleet een deelklasse is van NP. Maar waarom is deze klasse belangrijk voor de vraag of $P = NP$? Stel: we kunnen van een NP-compleet probleem Q laten zien dat het in P zit. Dan volgt dat $P = NP$, want voor een willekeurig NP-probleem R hebben we een polynomiaal algoritme dat een vraag met betrekking tot probleem R omzet naar een vraag in probleem Q. Die vraag van Q is in polynomiale tijd op te lossen, omdat we aannamen dat $Q \in P$. De samenstelling van de reductie en Q is wederom een polynomiaal algoritme. Dus hebben we een polynomiaal algoritme om probleem R op te lossen.

Er zijn zowel positieve als negatieve gevolgen als blijkt dat $P = NP$. Een positief gevolg is dat veelvoorkomende NP(-complete) problemen toch 'snel' blijken te zijn. Een bekend voorbeeld is Travelling Salesman Problem, de vraag wat de korste route is tussen n steden. Verder zijn veel algoritmen die worden gebruikt in operations research en integer programming NP-compleet.

Dit gevolg is eigenlijk ook een negatief gevolg. Er zijn problemen die

nu nog 'moeilijk' zijn waarvan we willen dat dat zo blijft. Bijvoorbeeld het ontbinden van een natuurlijk getal in priemfactoren en het discrete logaritme probleem. De cryptografie heeft bijvoorbeeld erg veel baat bij deze 'moeilijke' problemen. Zo zijn alle public-key cryptosystemen gebaseerd op dit soort problemen (zie hoofdstuk 6). Zo ook zijn internetbetalingen en transacties veilig omdat er een 'moeilijk' probleem is dat het beschermd. Wanneer dit allemaal 'snelle' algoritmen blijken te zijn moeten al deze systemen worden gewijzigd of vervangen. Een tussenpersoon zou anders eenvoudig betalingen kunnen aanpassen.

Er is nog een verschil in het soort bewijs dat kan worden gevonden. Als er een constructief bewijs wordt gevonden, dan hebben we een direct probleem. Dan zijn direct alle 'moeilijke' problemen 'makkelijk'. Wanneer er echter een niet-constructief bewijs het licht ziet, dan weten we wel dat deze problemen 'snel' zijn, maar een 'snel' algoritme moet dan eerst nog gevonden worden voordat het problemen oplevert.

5.3 Een doorbraak

Hendrik Lenstra, Arjen Lenstra en Lázsló Lovász hebben hun algoritme gepubliceerd in een artikel genaamd *Factoring Polynomials with Rational Coefficients* [4] in 1982. In dit artikel wordt het algoritme gebruikt voor polynoomfactorisatie over $\mathbb{Q}$. Dit artikel was een grote doorbraak, want het algoritme voor polynoomfactorisatie, waarvan het LLL-algoritme dus een onderdeel is, was een *polynomiaal algoritme*. Terwijl wiskundigen over de hele wereld er vanuit gingen dat zo'n algoritme niet bestond. Iedereen had het sterke vermoeden dat polynoomfactorisatie, in ieder geval over $\mathbb{Q}$, geen P-probleem was. Mensen waren zelfs bezig met het zoeken naar worst-case scenario's. Het vinden van een polynomiaal algoritme voor een probleem dat verwacht werd niet in P te zitten is natuurlijk één punt voor $P = NP$, in het nadeel van $P \neq NP$.

Bijna alle experts op het gebied van complexiteitstheorie geloven dat $P \neq NP$. Dit blijkt uit een poll van William Gasarch, professor in de informatica aan de universiteit van Maryland[3]. Hierin zijn experts gevraagd naar hun mening over $P = NP$ en wat ze verwachten van het bewijs. Veel theoretisch informatici zien weinig hoop, aangezien er in de afgelopen dertig jaar ook geen vooruitgang is geboekt. Tevens bestaat het vermoeden dat deze vraag onafhankelijk is van ZFC, de axioma's van Zermelo-Fraenkel en het keuze-axioma. Dit zou de vraag of $P = NP$ een gelijke status geven als

de continuümhypothese. Een andere kijk op de zaak is dat er nooit een bewijs zal worden gevonden: "Als een eenvoudige vraag zoals de laatste stelling van Fermat al zo'n lang bewijs heeft, dan zijn er zeker ook vragen waarvan het bewijs te lang is om te vinden." László Lovász heeft er het volgende over gezegd: "Probably some new math modeling the information flow through a boolean circuit. With luck, something like algebraic topology or algebraic geometry will be used." Waar de meeste experts het over eens zijn is dat er sowieso onderzoek moet worden gedaan naar zowel de hypothese $P = NP$ als naar de hypothese $P \neq NP$.

6 Toepassingen van het LLL-algoritme

Waarom zouden we een case study doen over het LLL-algoritme? Er zijn genoeg algoritmen om uit te kiezen. Waarom geen case study over het algoritme van Euclides of het quick-sortalgoritme? Een belangrijke reden is het aantal toepassingen van het LLL-algoritme. Sinds de publicatie van het algoritme duikt het regelmatig op. Daarnaast zijn er bij het ontstaan van het algoritme veel toevalligheden betrokken. Zo is er de toevallig wijze waarop Hendrik Lenstra op basisreductie uitkwam, via Van Emde Boas. Daarnaast is er Lovász die de lezing van Lenstra kon gebruiken voor zijn eigen onderzoek. Ook hebben we Arjen Lenstra die op het juiste moment bezig was met zijn masterscriptie over polynoomfactorisatie.

Het artikel waarin het LLL-algoritme is gepubliceerd, had in de titel 'polynoomfactorisatie' en niet 'basisreductie'. De auteurs hebben hiervoor gekozen omdat de factorisatie juist het bijzondere resultaat is, niet de reductie. Of zoals Hendrik Lenstra in ons interview verwoordde: "Niemand wordt geil van een korte vector in een rooster." De eerste belangrijke toepassing van het algoritme is dus polynoomfactorisatie. Deze toepassing is niet erg praktisch. De toepassing zelf is een zuiver wetenschappelijk onderwerp: polynoomfactorisatie wordt in 'de praktijk' niet gebruikt. Andere toepassingen van het LLL-algoritme zijn ook al eerder aan bod gekomen. Zo is er ook de ellipsoïdemethode waarvoor Lovász het algoritme nodig had. Tenslotte is er het geheeltallig programmeren, de oorspronkelijke toepassing van Hendrik Lenstra.

Er zijn ook praktische toepassingen van het LLL-algoritme. Zo kan het worden gebruikt bij het converteren van JPEG-plaatjes om de juiste kleurenruimte te vinden. Ook kan het worden toegepast in draadloze netwerken met meerdere zenders en ontvangers. We zullen nu echter diep in gaan op één specifieke toepassing van het LLL-algoritme in de cryptografie. We bekijken een cryptosysteem gebaseerd op het knapzakprobleem en laten met behulp van het LLL-algoritme zien dat dit systeem onveilig is. Deze toepassing kwam vrij snel na de publicatie van het algoritme, namelijk in 1984 door A. Shamir[7].

6.1 Het knapzakprobleem en cryptografie

Het knapzakprobleem is een bekend probleem in het lineair programmeren. We zijn hier echter enkel geinteresseerd in een toepassing van het probleem in

de cryptografie. Het probleem is als volgt: We hebben een knapzak en willen deze vullen met blikken voedsel. Ieder blik heeft ook een bepaalde inhoud en een bepaalde voedingswaarde. Nu is de vraag: "Welke blikken nemen we mee in de knapzak, op zo'n manier dat zo veel mogelijk voedingswaarde wordt meegenomen?"

Een speciaal geval hiervan is wanneer de inhoud gelijk is aan de voedingswaarde. Dan wordt de vraag: "Welke deelverzameling van blikken geven de inhoud van de knapzak als som?" Dit is bekend als het deelverzameling-som-probleem. Het is goed mogelijk dat er niet zo'n deelverzameling bestaat en soms is de verzameling te groot om snel de optimale oplossing te vinden. Het deelverzameling-som-probleem is, net als het algemene knapzakprobleem, NP-compleet. Er is echter een speciaal geval waarvoor dit probleem eenvoudig is op te lossen, namelijk wanneer de elementen van de verzameling een *superstijgende* rij vormen.

Definitie: Een eindige rij $r = (r_1, r_2, ...r_n)$ met $r_i \in \mathbb{N}$ heet een *superstijgende rij* als voor alle $i \in \{2, 3, ...n\}$ geldt dat $r_i > \sum_{j=1}^{i-1} r_j$.

Zij $r = (r_1, r_2, ...r_n)$ een superstijgende rij en zij m een natuurlijk getal. De vraag of er een deelverzameling van r is waarvan de som m is, is nu eenvoudig op te lossen. Neem de grootste r_i zodanig dat $r_i < m$. Dit getal is zeker nodig voor de gewenste deelverzameling, want het is groter dan de som van de voorgaande getallen en de daarop volgende getallen zijn allemaal te groot. Neem nu de grootste r_j kleiner dan $m - r_i$, enzovoorts. Dit proces eindigt wanneer we een getal overhouden kleiner dan r_1 of als r_1 in de som zit. Als we nul overhouden, hebben we de gewenste deelverzamling gevonden. Houden we een getal groter dan nul over, dan hebben we in ieder geval de maximale deelverzameling-som kleiner dan m gevonden.

In 1978 werd een cryptosysteem ontwikkeld door Merkle en Hellman[6]. Dit systeem is gebaseerd op het knapzakprobleem, meer specifiek, op het deelverzameling-som-probleem. Dit was een van de eerste *public key* cryptosystemen. Een cryptosysteem is een systeem om 'veilig' boodschappen te kunnen verzenden en ontvangen. Veilig betekent in dit geval: zonder dat een derde persoon de boodschap kan lezen. Een public key cryptosysteem is een systeem waarbij degene die de boodschappen ontvangt, een openbare sleutel heeft die iedereen kan zien. Die sleutel wordt gebruikt om de boodschap te coderen. De gecodeerde boodschap wordt vervolgens

verzonden, waarna de ontvanger de boodschap kan decoderen met behulp van zijn privésleutel. Deze privésleutel is, zoals de naam doet vermoeden, alleen bekend bij de ontvanger en met behulp van deze sleutel is decoderen eenvoudig. Nu is een public key cryptosysteem veilig wanneer het niet mogelijk is om snel, dat wil zeggen in polynomiale tijd, een gecodeerd bericht te decoderen *zonder* kennis van de privésleutel.

Het RSA-cryptosysteem, momenteel het meestgebruikte cryptosysteem, is gebaseerd op het feit dat het moeilijk is om een getal te ontbinden in priemgetallen. Het cryptosysteem gebaseerd op het deelverzameling-som-probleem is niet in gebruik want het is geen veilig systeem. Zoals verderop in dit hoofdstuk zal blijken kan het LLL-algoritme worden gebruikt om dit aan te tonen. We zullen nu eerst bekijken hoe dit cryptosysteem werkt, waarna we zullen aantonen dat het is te kraken met behulp van het LLL-algoritme.

6.2 Een cryptosysteem

Voor het knapzakcryptosysteem zijn de volgende dingen vereist:

- Een superstijgende rij $w = (w_1, w_2, ...w_n)$.
- Een $q > \sum_{i=1}^{n} w_i$.
- Een r zo dat $\text{ggd}(r, q) = 1$.

Bereken vervolgens eerst $b = (b_1, b_2, ...b_n)$, waar $b_i \equiv rw_i \pmod{q}$. Merk op dat b in het algemeen geen superstijgende rij is.

b is nu de *openbare* sleutel en de *privé*sleutel is (w, r, q). Nu wil A(lice) een bericht $a = (a_1, a_2, ...a_n)$ sturen naar B(ob). Het bericht is in het eenvoudigste geval een stuk tekst, maar het kan ook een bestand of een plaatje zijn. De boodschap wordt in ieder geval binair geschreven. Dat wil zeggen dat voor alle i a_i enkel een 0 of een 1 kan zijn. Hoe de boodschap wordt omgezet tot binaire code is voor dit verhaal niet relevant.

Voordat A de boodschap verzendt, wordt deze met behulp van de openbare sleutel b tot $c = \sum_{i=1}^{n} a_i b_i$ gecodeerd. Daarna zendt A c naar B. a heet in de cryptografie de *plaintext* en c wordt de *cyphertext* genoemd. Merk hierbij op dat we nu te maken hebben met een

deelverzameling-som-probleem, want c is de som van een deelverzameling van b. Dit komt precies omdat alle a_i's 0 of 1 zijn.

B kent de privésleutel en kan daarmee c ontcijferen, zodat hij de originele boodschap, de plaintext a, weer terugkrijgt. Kortom, B weet c en zoekt $a = (a_1, a_2, ...a_n)$ zo dat $c = \sum_{i=1}^{n} a_i b_i$. Het decoderen gaat op de volgende manier:

B berekent $c' \equiv cr^{-1} \pmod{q}$. Dan volgt dat

$$\begin{aligned} c' \equiv cr^{-1} &\equiv \sum_{i=1}^{n} a_i b_i r^{-1} \pmod{q} \\ &\equiv \sum_{i=1}^{n} a_i w_i r r^{-1} \pmod{q} \\ &\equiv \sum_{i=1}^{n} a_i w_i \pmod{q} \end{aligned}$$

Maar dit is een eenvoudig geval van een deelverzameling-som-probleem, omdat w een superstijgende rij is. Merk op dat r^{-1} bestaat, omdat $\text{ggd}(r, q) = 1$, en tevens dat $c' < q$, omdat $\sum w_i < q$.

6.3 Het systeem gekraakt

Zoals al eerder is gezegd, is dit systeem niet veilig. Iemand die enkel de gecodeerde boodschap en de openbare sleutel kent kan ook snel, dat wil zeggen in polynomiale tijd, de originele boodschap terugvinden. Een kraker kan op de volgende manier te werk gaan.

Met enkel kennis van c en $b = (b_1, b_2, ...b_n)$ wordt de volgende matrix M gemaakt:

$$M = \begin{pmatrix} I_{n\times n} & 0_{1\times n} \\ b & -c \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ b_1 & b_2 & \dots & b_n & -c \end{pmatrix}$$

Deze matrix kunnen we zien als $n + 1$ kolomvectoren die een rooster opspannen. Passen we op dit rooster het LLL-algoritme toe, dan komen

er korte vectoren uit. Het enige wat het algoritme doet is vectoren bij elkaar optellen en van elkaar aftrekken en verwisselen. Om dus de $-c$ in matrix M klein te krijgen, telt het algoritme er andere basisvectoren bij op. Hierdoor veranderen de bovenste n rijen van de matrix natuurlijk ook. De lege kolom die boven $-c$ is geplaatst houdt bij welke waarden er bij $-c$ zijn opgeteld. Voor iedere keer dat b_i bij c is opgeteld, wordt de waarde van de i^e rij in de lege kolom met één verhoogd. Nu is het zo dat het zeker is dat een deelverzameling van b als som c oplevert. Omdat het algoritme de vectoren zo klein mogelijk probeert te maken, zal de $-c$ uiteindelijk 0 worden. In die kolom staan dan enkel nullen en enen, namelijk één 1 voor iedere b_i die er nodig was om $-c$ nul te maken. De oplossing die de kraker zoekt, staat dus in die kolom die enkel nullen en enen bevat. Die kolom zal de originele boodschap a zijn. Kortom: iedereen kan zonder privésleutel de cyphertext decoderen.

Natuurlijk kan ook zonder LLL-algoritme iedereen de originele boodschap terugvinden: men loopt gewoon alle mogelijkheden af. Het probleem is echter dat er dan 2^n mogelijkheden gecontroleerd moeten worden. Dat kost erg veel rekentijd: het is NP. Het LLL-algoritme is wel een polynomiaal algoritme, dus een kraker is met bovenstaande methode 'weinig' tijd kwijt met het vinden van de originele boodschap. We zien hieraan dat het knapzakcryptosysteem niet veilig is. Er zijn varianten van dit cryptosysteem bedacht, waaronder een waarbij de b worden gemaakt door herhaalde vermenigvuldiging. Bijna al deze systemen blijken niet veilig te zijn. Natuurlijk is geen enkel systeem dat momenteel bestaat veilig als blijkt dat $P = NP$.

Zoals al eerder gezegd, is het knapzakprobleem een NP-compleet probleem. Maar hierboven staat een methode om in polynomiale tijd een oplossing te vinden voor een deelverzameling-som-probleem. Is dit dan toch P? Het antwoord is nee. Dit is niet waar omdat we hier in een speciaal geval zitten. We weten namelijk dat er een deelverzameling is die als som precies c heeft, zo is c geconstrueerd. In het algemeen is de vraag: "Is er een deelverzameling met als som een gegeven waarde?" of "Zoek de maximale som van deelverzamelingen kleiner dan een gegeven waarde." In dat geval is het, vooralsnog, niet mogelijk om een antwoord te geven in deterministische polynomiale tijd.

7 Wiskundigen in het kwadrant van Pasteur

In dit hoofdstuk vragen we ons af of we reden hebben te denken, afgezien van het LLL-algoritme, dat het kwadrant van Pasteur in de wiskunde bestaat. Zijn niet alle wiskundigen, in tegenstelling tot andere exacte wetenschappers, altijd in te delen als 'puur theoretisch onderzoek' en 'puur toegepast onderzoek'? Dat is maar zeer de vraag. In dit hoofdstuk zullen we voorbeelden bespreken van wiskundigen en wiskundig onderzoek die een veel duidelijker toepassingsgericht karakter hebben. In veel voorbeelden gaat het ofwel om onderzoekers die direct of indirect bij het LLL-algoritme betrokken waren, ofwel om onderzoekers die Hendrik Lenstra of Ionica Smeets noemde als mogelijke voorbeelden van 'wiskundige Pasteurs'. Doen zij inderdaad onderzoek met een concreet, maatschappelijk toepassingsdoel? En, wellicht meer voor discussie vatbaar: mogen wij van dit onderzoek óók fundamenteel nieuwe wiskundige kennis verwachten?

De hieronder genoemde gebieden zijn een selectie. In het bijzonder lijken ook de stochastiek en statistiek, die wij niet zullen bespreken, typische voorbeelden van werk in het kwadrant van Pasteur.

7.1 Operations research en combinatoriek

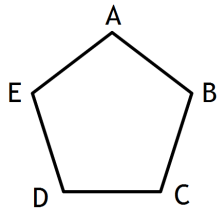
Van alle toegepaste wiskunde is het directe economische nut van operations research misschien wel het duidelijkst. Deze tak van de wiskunde houdt zich bezig met het zoeken van optimale of bijna optimale oplossingen van ingewikkelde beslisproblemen[10]. Typische toepassingen liggen in sectoren als de logistiek, maakindustrie (denk aan voorraadbeheer), telecommunicatie en zelfs dienstverlening (denk aan het maken van personeelsroosters). In vrijwel alle gevallen gaat het om het vinden van de voordeligste strategie voor allerlei micro-economische keuzes. Wat is de kortste route die langs alle klanten van vandaag leidt? Wat zijn de grootste bottlenecks in het internetverkeer en hoe lossen we die het efficiëntst op? We bespreken hier voorbeelden van wiskundig werk binnen de vaak gerelateerde gebieden van operations research, optimalisatie, die meestal tot de operations research wordt gerekend, en combinatoriek, met name de grafentheorie.

De grenzen zijn overigens zeker niet precies afgebakend. Het is opvallend hoe vaak wij in ons onderzoek namen zijn tegengekomen die werken op een snijvlak van deze disciplines: de combinatorische optimalisatie. Dit is bij uitstek het gebied van László Lovász, die voor zijn werk tal van

prijzen in de wacht heeft gesleept, maar ook van bijvoorbeeld Alberto Marchetti-Spaccamela, die samen met Peter van Emde Boas Hendrik Lenstra, tot zijn aanvankelijke irritatie, op het spoor van de roosterbasisreductie zette. De bekendste Nederlandse naam in dit gebied is ongetwijfeld Lex Schrijver. Hij ontving in 2005 de Spinozapremie, de 'Nederlandse Nobelprijs' die in 1998 ook aan Hendrik Lenstra is toegekend (die overigens niet met combinatoriek maar vooral met de computationele getallenleer naam heeft gemaakt). Schrijver heeft, in de woorden van het juryrapport, "samenhang gebracht in de combinatorische optimalisering" en dit veld "naar een hoger niveau getild"[11]. Schrijver houdt zich niet alleen bezig met zuivere wiskunde: hij speelde een actieve rol in het opstellen van het *Spoorboekje 2007*, een samenkomst van theorie en praktijk die de andere projectgroep van het Modellenpracticum uitvoerig zal bespreken.

Hier zullen we proberen een idee te geven van dit vakgebied en zijn toepassingsmogelijkheden door nog iets verder in te gaan op het werk van Lovász. Men zou misschien verwachten dat combinatorische optimalisatie een vak is waarin concrete toepassingen voorop staan, maar dat is zeker niet altijd het geval. Vrijwel alle publicaties van Lovász hebben juist een zeer sterk theoretisch karakter, met titels als *The Geometry of Logconcave Functions and an* $O(n^3)$ *Sampling Algorithm*; *Reflection positivity, rank connectivity, and homomorphism of graphs*; *Finitely forcible graphons*; of *A Borsuk theorem for antipodal links and a spectral characterization of linklessly embeddable graphs*. Waar gaat zijn vak over, behalve grafentheorie? We beschouwen een niet al te esoterisch voorbeeld uit een van zijn bekendere[12] papers: *On the Shannon Capacity of a Graph* (1979)[5].

In dit artikel berekent Lovász de Shannoncapaciteit (de bandbreedte, in ict-taal) van communicatiekanalen die op een bepaalde manier met een graaf zijn te modelleren. Stel dat je telkens een kort elektrisch signaal door een kabel stuurt, met telkens gelijke amplitude en frequentie maar met steeds andere fase (ten opzichte van een zeker referentietijdstip). Deze fase is uit te drukken in een getal van 0 tot 2π, maar met dien verstande dat 0 weer heel sterk op 2π lijkt. Of stel dat je een signaal verstuurt door borden in de lucht te houden in felle regenboogkleuren. Het zijn dus kleuren uit de kleurencirkel, die je wederom zou kunnen beschrijven door een hoek in radialen modulo 2π. Neem nu aan dat de ontvanger de gemeten hoek discretiseert tot vijf waarden: A, B, C, D en E. Dit zijn de signalen waaruit de afzender zijn bericht heeft opgebouwd. Door meetonnauwkeurigheden op grote afstand gaat de datacommunicatie soms fout, maar nooit op een dramatische manier: men



Figuur 2: De vijfhoekgraaf. Lovász heeft de Shannoncapaciteit berekend van communicatiekanalen die deze graaf beschrijft.

verwart misschien een C met een B of een D, maar nooit met een A of een E. Vanwege het cyclische domein van het signaal is het ook mogelijk dat E en A verward worden. Je zou de situatie kunnen beschijven met de graaf in figuur 2. Hierin geeft elke knoop een mogelijke waarde van het signaal aan, en een kant geeft aan dat twee waarden mogelijk verward kunnen worden. Stel dat we één teken per milliseconde verzenden. Wat is dan de capaciteit van het datakanaal?

Een voor de hand liggende schatting zou zijn: één bit per milliseconde. Het maximale aantal gegarandeerd onderscheidbare tekens uit de graaf is immers twee: A en C, bijvoorbeeld, of B en E, maar niet A, C en E want dat levert al een mogelijke verwarring op tussen A en E. Per milliseconde zijn er dus twee mogelijkheden, oftewel ${}^2\log 2 = 1$ bit. Maar we kunnen iets slimmers verzinnen. We verdelen het hele signaal in blokken van twee. Bouw nu een bericht op uit de volgende codewoorden: AA, BC, CE, DB en ED. Deze vijf woorden zijn nooit te verwarren: AA niet met BC omdat de A en de C niet te verwarren zijn, BC niet met ED omdat de B en de E niet te verwarren zijn, et cetera. We kunnen slechts één codewoord verzenden per twee milliseconden, maar zo'n codewoord bevat wel ${}^2\log 5 \approx 2,322$ bits aan informatie. Per milliseconde is dat $\frac{1}{2}\,{}^2\log 5 = {}^2\log\sqrt{5} \approx 1,161$ bits: meer dus dan met de teken-voor-tekenmethode. De vraag is nu: wat is de maximale capaciteit van dit kanaal als we onze codewoorden willekeurig lang mogen maken (en dus geen rekening houden met het probleem van tijdelijke opslag van heel lange codewoorden, benodigde rekenkracht, et cetera)? Lovász kon bewijzen dat voor de vijfhoek een bitrate van ${}^2\log\sqrt{5}$, die we al met codewoorden van slechts twee signalen konden bereiken, ook maximaal is. In dit voorbeeld bereikten we die capaciteit toevallig al met codewoorden van slechts twee signalen, maar het is in principe ook mogelijk dat de maximale capaciteit alleen maar asymptotisch te bereiken is door

steeds langere codewoorden te gebruiken. In het algemeen is het vraagstuk zeer moeilijk oplosbaar. Lovász vond een functie waarmee een bovenschatting van de Shannoncapaciteit van willekeurige grafen te berekenen is.

Dit uitgebreide voorbeeld toont aan dat ogenschijnlijk simpele vragen uit de informatietheorie en informatica heel ingewikkelde en diepzinnige antwoorden kunnen vergen. Vaak zijn methoden uit de combinatoriek en combinatorische optimalisatie nodig, en, zoals het werk van Lovász en vele anderen laat zien, is het ook een belangrijke inspiratiebron voor wezenlijk nieuwe inzichten in die vakgebieden. Zo geldt het probleem van de Shannoncapaciteit van grafen nog steeds als zeer moeilijk[13]. Tegelijkertijd ligt een belangrijk toepassingsgebied voor de hand: de datacommunicatie. Kunnen we daarom dit soort werk als een typisch voorbeeld voor het kwadrant van Pasteur zien? Zoals het combinatorische werk van Schrijver heeft laten zien zal dat in veel gevallen heel redelijk zijn. Maar in onze ogen is het bovenstaande voorbeeld uit het werk van Lovász toch net iets meer 'Bohr' dan 'Pasteur'.

Ten eerste heeft goed zoeken ons geen toepassingen opgeleverd waarin de Shannoncapaciteit van een graaf, dus de theoretische limiet die Lovász probeert te vinden, in praktische datacommunicatietoepassingen werd gebruikt. (Dit in tegenstelling tot het begrip Shannoncapaciteit in het algemeen, en de formule van Shannon-Hartley in het bijzonder, waarnaar een korte zoektocht direct heel toepassingsgericht werk oplevert[14].) De redenen liggen voor de hand: het berekenen van de theoretische limiet is voor een willekeurige graaf heel lastig, terwijl toepassers vooral geïnteresseerd zijn in de kanaalcapaciteit die met een enigszins redelijke woordlengte te behalen is. Niet alleen zijn die vaak met wat trial-and-error wel in goede benadering te vinden, vaak neemt de meeropbrengst van extra lange woorden heel snel af.

Ten tweede is het niet duidelijk dat Lovász dit werk uitvoerde met meer dan een vaag idee van mogelijke toepassingen. Hij rept daar in ieder geval met geen woord over in zijn artikel. Het is vrijwel zeker dat hij niet direct met toepassers contact heeft gehad. Deze aanpak past goed in het kwadrant van Bohr, of in het lineaire model: een moeilijk wiskundig probleem proberen op te lossen, misschien geïnspireerd maar zeker niet opgelegd door de echte wereld, en het aan anderen laten om daar al dan niet iets mee te doen.

7.2 Numerieke wiskunde

Het is nauwelijks een overdrijving om te stellen dat vrijwel alle problemen uit de natuurkunde uiteindelijk neerkomen op het oplossen van ofwel integralen ofwel (partiële) differentiaalvergelijkingen. In sommige gevallen, zoals de elektronenbanen van het waterstofatoom, is met pijn en moeite een exacte oplossing te vinden in termen van elementaire functies. Veel vaker komt het voor dat de oplossingen, als al kan worden aangetoond dat ze bestaan en goed gedefinieerd zijn, niet algebraïsch te vinden zijn. In zulke gevallen moeten numerieke methoden uitkomst bieden. De numerieke wiskunde probeert continue problemen op de een of andere manier te discretiseren om ze vervolgens in een goede benadering door computers op te laten lossen. Het praktische belang van zulke methoden is nauwelijks te overschatten. Zonder numerieke wiskunde waren stromingsleer, fysische chemie of theorie van gecondenseerde materie waarschijnlijk weinig meer dan fraaie intellectuele bouwsels zonder al te grote voorspellingskracht buiten de allersimpelste gevallen.

Een bekende naam in dit vakgebied is Alfio Quarteroni. Het was zijn naam die als eerste in het hoofd van Hendrik Lenstra opkwam toen wij hem vroegen of hij voorbeelden kon noemen van wiskundigen die hij tot het kwadrant van Pasteur zou rekenen. Deze Italiaan heeft vele bijdragen geleverd aan meer algemene methoden om allerlei problemen met numerieke methoden op te kunnen lossen. Je zou dit zeker fundamenteel relevant onderzoek kunnen noemen. Zo mocht Quarteroni in 2006 een plenaire lezing houden op het International Congress of Mathematicians, een zeer prestigieuze conferentie die eens in de vier jaar wordt gehouden en waar ooit Hilbert zijn beroemde lijst van 23 onopgeloste wiskundige problemen presenteerde. Hij zet deze kennis echter ook voortdurend in om allerlei heel concrete problemen aan te pakken, vooral op het gebied van de stromingsleer. Zo heeft hij gewerkt aan het modelleren van de menselijke bloedsomloop[15][16]. Dit soort onderzoek is niet alleen een heel pittige en wiskundig interessante testcase van en stimulans voor de numerieke methoden die in de stromingsleer gebruikt worden, maar het wordt bovendien uitgevoerd met de hoop dat het in de toekomst leidt tot beter begrip van, en misschien wel betere behandelmethoden voor, medische problemen als aderverkalking[17].

Maar de toepassingen kunnen nog veel concreter. In 2003 hielp Quarteroni met zijn vloeistofdynamische simulaties het Zwitserse team eerste

te worden in America's Cup, de bekendste zeilwedstrijd ter wereld[18], een overwinning die nog nooit door een Europees team in de wacht was gesleept. Het bleef niet bij één in het oog springend project: van 2003 tot 2007 was hij actief betrokken bij het plan van de eveneens Zwitserse avonturier Bertrand Piccard om zonder tussenstops de wereld rond te vliegen in een volledig op zonne-energie werkend vliegtuig[19]. Of dit soort werk nog een zuiver wiskundige relevantie kan worden toegedicht is natuurlijk de vraag.

7.3 Cryptografie

We komen nog even terug op de cryptografie. In zekere zin is dit zowel het oudste als het modernste toepassingsgebied van de wiskunde. Julius Caesar gebruikte al een eenvoudig 'substitutiecijfer' om te voorkomen dat de vijand de berichten aan zijn generaals zou kunnen onderscheppen:

> *If he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely D, for A, and so with the others.*
> −Suetonius, Life of Julius Caesar[20]

Het is heel eenvoudig om geheimschrift van dit type te ontcijferen, maar in de tijd van Caesar was het zeker beter dan niets. (Er zijn overigens antieke bronnen die suggereren dat Caesar ook complexere vormen van versleuteling gebruikte.) In de Tweede Wereldoorlog lag dat wel anders. Genoodzaakt door de gigantische schaal van de militaire operaties en het enorm toegenomen belang van radiocommunicatie, die door iedereen is af te luisteren, bereikte de cryptografie nieuwe hoogten. De Duitse strijdmacht gebruikte ingenieuze versleutelingskastjes, de Enigma-machines, nota bene uitgevonden (of althans gepatenteerd) door een Nederlander, die wel een behoorlijk grote mate van cryptografische veiligheid boden[21].

Terwijl iedereen in het geallieerde kamp de hoop had opgegeven op het ontcijferen van de Enigma-machine die de Duitse marine in gebruik had, wist een jonge wiskundige, Alan Turing, het probleem toch op te lossen[22]. In Bletchley Park, het zenuwcentrum van ontcijferaars dat in allerijl was opgetrokken uit de Britse wiskundige elite van die tijd, ontwikkelden Turing en anderen een machine, de 'Bombe', die in hoog tempo de brute rekenkracht kon leveren die toch nog nodig was om de sleutels

van de vijand te kunnen vinden. De vaardigheid om de radioberichten van de Duitsers te ontcijferen, met name die van en naar de gevreesde U-boten, leverde de geallieerden een enorm strategisch voordeel op. Maar Turing heeft ook enorme bijdragen geleverd aan de theorie, zelfs aan de grondslagen van de wiskunde. Aan de hand van zijn concept van de 'universele machine', nu beter bekend als de Turingmachine, bewees hij dat het onmogelijk is een programma te schrijven dat vooraf kan bepalen of een willekeurig gegeven algoritme zijn berekening in eindige tijd kan afronden, een stelling die equivalent is met de onvolledigheidsstelling van Gödel. Het is aannemelijk is dat tussen het theoretische en praktische werk van Turing een duidelijke kruisbestuiving heeft plaatsgevonden. Opmerkelijk genoeg voor een wiskundige van Cambridge probeerde Turing steeds zijn theoretische ideeën met werkende apparaten in de praktijk te brengen. Helaas is het hem nooit gelukt om voor zijn hartenwens, de bouw van een échte 'universele machine', de handen op elkaar te krijgen, maar zijn visionaire blik is overduidelijk nu zijn universele machine ons overal omringt. Bij Alan Turing waren het zuivere en het praktische onmiskenbaar in één persoon verenigd.

Al deze ontwikkelingen dateren van voor de grote opkomst van operations research of de computertechniek en informatica, die numerieke methoden voor veel complexere problemen bruikbaar maakten (al liet onder meer het Mathematisch Centrum, nu CWI, ook eerder wel grote numeriek wiskundige problemen doorrekenen door ijverige dames, grotendeels met de hand[23]). Toch is cryptografie ook een heel moderne ontwikkeling. Door de opkomst van de personal computer en vooral het internet is goede cryptografie, in het bijzonder public-key cryptography, in de afgelopen tien, vijftien jaar een fenomeen geworden waarmee vrijwel iedereen dagelijks te maken heeft en waarvan steeds meer zakelijke transacties afhankelijk zijn.

De wiskunde van veiligheid op internet, diep weggestopt achter webbrowsers en *Secure Socket Layers*, laat soms onverwacht haar gezicht zien. In 2005 toonden Arjen Lenstra, Benne de Weger en Xiaoyun Wang aan dat het niet alleen in theorie maar ook praktisch mogelijk was twee verschillende websitebeveiligingscertificaten (X.509) te construeren met dezelfde MD5-hash ('handtekening'), een zogenaamde *hash collision*. Dit betekent dat deze cryptografische methode niet meer veilig is. Gelukkig gebruikten webservers toen al meestal het veiligere SHA-1-algoritme, maar ook dat begint nu haarscheurtjes te vertonen.

De praktische relevantie van cryptografisch werk is heel groot. Maar valt het daarmee ook in het kwadrant van Pasteur? Dat zal zeker niet altijd het geval zijn. Wiskundig is het verschil misschien niet zo interessant tussen een hash collision van twee willekeurige stukjes data, een resultaat dat voor MD5 al bekend was, en een collision in de hash-functies van twee beveiligingscertificaten van een gegeven type, de bijdrage van Arjen Lenstra cum suis. Maar voor wezenlijke doorbraken in de cryptografie, evengoed als voor werk aan de complexiteitsklassen van bepaalde algoritmen, zijn vaak wezenlijk nieuwe wiskundige inzichten nodig. Zoals Hendrik Lenstra aangaf tijdens ons interview: altijd bestaat de mogelijkheid dat cryptografische problemen waarvan je hoopt dat ze te moeilijk zijn om te kraken toch simpel blijken. "Als een cryptosysteem het wiskundegebied X gebruikt, dan is de enige manier om het te breken of te onderzoeken: meer leren over gebied X." Terug naar de zuivere wiskunde dus!

8 Conclusie

Op het eerste gezicht lijkt het LLL-algoritme duidelijk te passen in het kwadrant van Pasteur. Er zijn vele, vaak praktische, problemen die met het LLL-algoritme zijn opgelost. Dit verslag geeft daarvan een uitgewerkt voorbeeld (zie paragraaf 6.1). Het is echter belangrijk om op te merken dat Stokes' kwadrantenmodel niet zozeer draait om de vraag of het bewuste onderzoek leidt tot praktische toepassingen als om de vraag met welke intentie het onderzoek aanvankelijk werd uitgevoerd. Ook het atoommodel van Bohr heeft later immers enorm veel toepassingen gevonden. Wij geloven dat, zo bezien, de ontwikkeling van het LLL-algoritme verreweg het beste geschaard kan worden onder het 'zuiver wetenschappelijk onderzoek' (Bush) dan wel 'kennisgericht maar niet direct toepassingsgericht onderzoek' (Stokes): het kwadrant van Bohr.

Waaruit blijkt dat de onderzoekers die bijdroegen aan het LLL-algoritme hoofdzakelijk gericht waren op het vergaren van fundamentele kennis? De motivatie van iedere onderzoeker is natuurlijk anders, maar wat we over elk van de betrokkenen weten wijst meestal in dezelfde richting. Van Hendrik Lenstra hebben wij informatie uit eerste hand.

In Lenstra's visie leiden fundamentele wiskundige doorbraken in stappen tot praktische toepassingen: de fundamentele onderzoekers komen met een geheel nieuw resultaat (zeg: 'probleem X is in polynomiale tijd oplosbaar'); iets meer toepassingsgerichte onderzoekers gaan dit resultaat fine-tunen en verbeteren (een algoritme in P met een complexiteitsveelterm van lagere graad of met lagere coëfficiënten vinden); weer een ander ziet dat dit algoritme een bepaald praktisch probleem in principe zou kunnen oplossen, et cetera. Uiteindelijk komt de vinding zo terecht bij de "mensen in blauwe overalls" die haar kunnen toepassen in een betaalautomaat of een fabriek. Deze visie op het proces van innovatie lijkt sterk op het lineaire model, met name op het aspect van het lineaire model dat Stokes de 'dynamische vorm' noemt.

Fundamentele motivatie

In het interview dat wij Hendrik Lenstra hebben mogen afnemen geeft hij herhaaldelijk aan dat zijn grote passie lag en ligt bij het vooruithelpen van de wiskunde als zodanig, niet in het inzetten van die wiskunde voor iets anders. Hoewel hij toepassingen die voortkomen uit werk als het LLL-algoritme met

interesse volgt, laat hij het graag aan anderen om die toepassingen te vinden: "Ik heb een stelling en geef hem door." Hij laat er geen onduidelijkheid over bestaan dat Lenstra zichzelf helemaal aan het begin zou plaatsen van het stappenschema dat uiteindelijk tot technologische innovatie leidt: "Na één stap voel ik al dat ik er niet thuis ben."

Ook de andere betrokken onderzoekers hadden waarschijnlijk een duidelijk op fundamentele kennis gerichte motivatie. Zowel Alberto Marchetti-Spaccamela als Peter van Emde Boas, die Hendrik Lenstra voor het eerst, bijna tegen wil en dank, op het spoor van het roosterbasisreductieprobleem zette, hield zich onder meer bezig met theoretische informatica. Hoewel dit vakgebied naar wiskundige maatstaven 'toegepast' is – zo houdt het zich bezig met de complexiteit van algoritmen, een kwestie die van groot belang is bij het ontwikkelen van efficiënte computerprogramma's – is de insteek van veel theoretisch informatici, waaronder volgens Lenstra zeker ook die van Van Emde Boas en Marchetti, heel 'zuiver' van aard.

En hoe zit het met die andere twee L's, László Lovász en Arjen Lenstra? Ook zij zijn of waren betrokken bij problemen in de complexiteitstheorie en algoritmiek. Arjen Lenstra was van de drie de auteur die zich al langer had beziggehouden met de ontbinding van geheeltallige veeltermen. Hoewel het hier gaat om een in zekere zin 'praktische' vraag – hoe veel rekentijd is er nodig om zo'n veelterm te ontbinden? – had dit probleem toen, voor zover ons bekend, geen voorziene toepassing. Ook Arjen Lenstra was dus vermoedelijk vooral gemotiveerd door de hoop het begrip van fundamentele wiskundige problemen te vergroten.

Lovász' voornaamste onderwerp, ten slotte, was (en is) de combinatorische optimalisatie, wederom een vakgebied waarin het vinden van efficiënte algoritmen een belangrijke plaats inneemt. Hij hield en houdt zich daarbinnen veel bezig met grafentheorie. Dit is een vakgebied waarin mogelijke praktische toepassingen vaak voor de hand liggen. Het Travelling Salesman Problem is daarvan het bekendste voorbeeld. Desalniettemin maken ook de meeste publicaties van Lovász een sterk theoretische indruk. Je zou dit soort werk daardoor zowel in het kwadrant van Bohr als in het kwadrant van Pasteur in kunnen delen. Cruciaal is hierbij wiens motivaties je als uitgangspunt neemt. Het is goed mogelijk dat subsidieverstrekkers de hoop op praktische toepassingen die Lovász' vakgebied wekt meewegen in hun beslissing subsidie toe te kennen aan zijn onderzoek. Zo bezien zou het onderzoek in het kwadrant van Pasteur moeten worden ingedeeld.

Pasteur was echter, anders dan Lovász voor zover ons bekend, ook zelf direct betrokken bij de toepassingen van zijn werk. Hij deed veel van zijn inspiratie voor zijn theorie over micro-organismen op in brouwerijen en wijnbedrijven. Bovendien kon hij deze bedrijven, op basis van zijn nieuwe, fundamentele theorieën, ook direct adviezen geven om het fermentatieproces te verbeteren. Van een dergelijke directe betrokkenheid is in het werk van Lovász geen sprake. Als we uitsluitend Lovász' eigen motivatie en werkzaamheden in ogenschouw nemen, zouden we zijn werk dus duidelijk in het kwadrant van Bohr moeten indelen of, binnen het lineaire model, onder het zuiver wetenschappelijk onderzoek.

Verkooppraatje

We hebben gezien dat alle hoofdrolspelers in de ontdekking van het LLL-algoritme, althans als het gaat om hen persoonlijk, in het algemeen een duidelijk fundamentele, niet direct toepassingsgerichte motivatie hadden met het werk waarmee zij rond de publicatie van het artikel (1982) bezig waren. Maar er zijn meer en meer specifieke redenen om dit werk in het kwadrant van Bohr in te delen.

Allereerst is daar de naam van het bewuste artikel: *Factoring polynomials with rational coefficients*. Deze titel is enigszins opmerkelijk. Het is heel verdedigbaar om te stellen dat het LLL-algoritme een veel grotere impact heeft gehad dan het ontbindingsalgoritme, waarvoor het artikel het enkel als 'hulpmiddeltje' presenteert. (Dit is iets gechargeerd: het artikel besteedt ook ruim één pagina aan twee kleinere toepassingen op de Diophantische benadering.) Het LLL-algoritme heeft, zoals in hoofdstuk 6 is beschreven, inmiddels zeer veel toepassingen gevonden, zowel in de zuivere wiskunde (voor een selectie van toepassingen in de getaltheorie, zie [9]) als voor heel praktische problemen (het knapzakcryptosysteem, hercompressie van JPEG-afbeeldingen, ...) Het boek dat bij de conferentie ter gelegenheid van de 25e verjaardag van het artikel werd gepubliceerd (LLL+25) heet dan ook *The LLL Algorithm: Survey and Applications*[1]: ontbinding van rationale veeltermen is slechts één van de vele hoofdstukken.

Waarom noemden Lenstra, Lenstra en Lovász hun artikel dan toch naar die specifieke toepassing op ontbinding, in plaats van iets als 'een roosterbasisreductiealgoritme met toepassingen in de getaltheorie'? Hendrik Lenstra en Ionica Smeets leggen beide uit dat in die tijd vrijwel alle wiskundigen die zich met dit onderwerp bezighielden ervan uitgingen dat

het ontbinden van veeltermen met rationale coëfficiënten niet in polynomiale tijd te doen was. De redenering was ongeveer: het vinden van de priemfactorontbinding van gehele getallen is niet polynomiaal (dat wil zeggen: niet in $L = {}^2\log(n)$, het aantal bits van het ingevoerde getal n), dus het ontbinden van hele veeltermen van gehele getallen of breuken moet dan zeker ook niet polynomiaal zijn. Het werd zelfs als een zinvolle bezigheid gezien om pathologische gevallen te bedenken waarop bestaande ontbindingsalgoritmes hun tanden zeker stuk zouden bijten en zodoende aan te tonen dat die algoritmes in het algemeen niet polynomiaal zijn. Het LLL-algoritme bewees dat een polynomiaal algoritme wel degelijk mogelijk was[4]. Dit was een opzienbarend resultaat. Voor 'de L's' was het dan ook natuurlijk om dit resultaat tot hoofdonderwerp van hun artikel te maken. Hoewel de auteurs hebben aangegeven dat zij min of meer bij toeval op hun grote ontdekking stuitten, suggereert hun keuze voor deze titel dat het vergroten van fundamentele wiskundige kennis meer hun doel was dan het rechtstreeks bijdragen aan praktische toepassingen, die het ontbinden van rationale veeltermen, voor zover Hendrik Lenstra bekend, immers niet had.

Het interview met Hendrik Lenstra biedt nog meer redenen om te denken dat, ook waar het gaat om dit specifieke project, het werk van de drie auteurs in het kwadrant van Bohr ingedeeld zou moeten worden. Hadden de auteurs, naast hun fundamentele interesse, niet ook een toepassingsgericht doel? Hendrik Lenstra zegt daarover: "Wij wisten niet, hoewel het niet verbaasde, dat het LLL-algoritme verdere toepassing zou hebben." Dat kan dus niet de motivatie voor hun werk zijn geweest![4] Gevraagd of hij destijds voor zijn onderzoek een (oneerbiedig gezegd) 'verkooppraatje' had voor subsidieverstrekkers en andere beleidsmakers – een vergezicht op een aansprekende praktische toepassing waartoe zijn werk zou kunnen leiden – antwoordde Hendrik Lenstra dat dit niet het geval was. Zoiets wordt alleen geaccepteerd als je vrij concrete redenen hebt om te denken dat je zulke ideeën ook zou kunnen waarmaken. Die had hij destijds kennelijk niet, wat er wederom sterk op wijst dat het onderzoek dat leidde tot LLL niet toepassingsgericht was.

[4]Merk opnieuw op dat het kwadrantenmodel alleen toepassingen meeweegt die door de onderzoekers zelf min of meer werden voorzien of beoogd: ook het atoommodel van Bohr kreeg immers later zeer veel toepassingen.

Wiskundige Pasteurs

Maar betekent dit dat het model van Stokes niet van toepassing is op de wiskunde? Ons lijkt van niet. In hoofdstuk 7 hebben we geprobeerd te laten zien dat er wel degelijk voorbeelden bestaan van wiskundigen die in het kwadrant van Pasteur geplaatst kunnen worden. We noemden onder meer Alan Turing, Alfio Quarteroni en Lex Schrijver. Onze casus leert ons alleen dat het LLL-algoritme geen argument is *voor* dit model: het past even goed in het model van Bush.

Referenties

[1] Nguyen, Phong Q.; Vallée, Brigitte, *The LLL Algorithm, Survey and Applications*, Springer Publishing Company, Incorporated, New York, NY, 2009.

[2] Donald E. Stokes, *Pasteur's Quadrant*, Basic Science and Technological Innovation, Brookings Institution Press, 1997.

[3] William I. Gasarch, "The P=?NP poll", (2002), SIGACT News 33 (2): 34–47.

[4] Lenstra, A. K.; Lenstra, H. W., Jr.; Lovász, L., "Factoring polynomials with rational coefficients." (1982) Mathematische Annalen 261 (4): 515–534.

[5] L. Lovász: "On the Shannon Capacity of a Graph", (1979) IEEE Trans. Inform. Theory 25 , 1–7.

[6] R. Merkle; N. Hellman, "Hiding Information and Signatures in Trapdoor Knapsacks", (1978) IEEE Trans. Information Theory, IT-24-5, September.

[7] A. Shamir, "A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem", (1984) IEEE Trans. Inform. Theory, IT-30, 699–704.

[8] http://www.nsf.gov/about/history/vbush1945.htm

[9] http://www.math.unicaen.fr/ simon/maths/lll25_Simon.pdf

[10] http://en.wikipedia.org/wiki/Operations_research

[11] http://www.nwo.nl/files.nsf/Pages/NWOP_6CXDGB/$file/2005JuryrapportSchrijver.htm

[12] http://www.inamori-f.or.jp/laureates/k26_b_laszlo/prf_e.html,
(Selected publications)

[13] http://www.imfm.si/preprinti/PDF/00681.pdf

[14] http://csi.usc.edu/Willner.NSF/pdf/stewart-personick-shannon.pdf

[15] http://de.wikipedia.org/wiki/Alfio_Quarteroni

[16] http://mathicse.epfl.ch/cmcs/publications.php3?query=collection:ARTICLE

[17] http://mathicse.epfl.ch/cmcs/NewResearch/vascular.php3

[18] http://de.wikipedia.org/wiki/America%E2%80%99s_Cup

[19] http://mathicse.epfl.ch/cmcs/AQ/CV_Alfio_Quarteroni.pdf

[20] http://en.wikipedia.org/wiki/Caesar_cipher

[21] http://nl.wikipedia.org/wiki/Hugo_Alexander_Koch

[22] http://www.turing.org.uk/philosophy/ex5.html

[23] http://www.kennislink.nl/publicaties/rekenmeisjes-en-rekentuig

A Interviews

A.1 Henkdrik Lenstra

Met dit interview wilden we een betere inkijk krijgen in de motivatie van Hendrik Lenstra, maar ook van de andere betrokken wiskundigen. We proberen met het interview ook een goed idee te krijgen over het LLL-algoritme in het lineaire en het kwadrantenmodel. Maar voordat we hier meteen over beginnen, begonnen we met het nog eens terugkijken naar de geschiedenis van het LLL-algoritme.

Het begon in eerste instantie met geheeltallig lineair programeren, omdat van Emde Boas en Marchetti bezig waren met de vraag over het vinden van een punt met geheeltallige coördinaten in een gegeven driehoek. We moeten dus gaan kijken naar de oorsprong van lineair programmeren. Lineair programmeren is eigenlijk uit toepassingen ontstaan. Er zijn namelijk veel problemen oplosbaar door ze te lineariseren. "De grote link van lineair programmeren met wiskunde is lineaire algebra, want simpel gezegd is lineair programmeren lineaire algebra met de gelijktekens vervangen door ongelijktekens," aldus Lenstra.

Arjen Lenstra was bezig met factorisatie van polynomen over algebraïsche getallichamen, waar hij er door toeval op uitkwam dat dit met behulp van het LLL-algoritme polynomiaal was. Arjen zat vooral in het vakgebied van algoritmen en complexiteit, dat toch wordt gezien als meer toegepast. Polynoomfactorisatie had eigenlijk alleen toepassingen 'binnen de muren'. Buiten de wiskunde had het vrijwel geen toepassingen.

Toen Hendrik Lenstra de vragen van Marchetti en van Emde Boas kreeg, had hij al meteen in zijn hoofd: 'basisreductie'. Zijn benaderingswijze destijds noemt Lenstra heel "naïef," maar zijn grote voordeel was dat hij de 'taal' van complexiteitsklassen beheerste. Hendrik was juist bezig in getaltheorie, waardoor het opvallend was dat ze juist naar hem kwamen met de vraag. Maar van Emde Boas was een goede vriend van Lenstra, waardoor het handig was om eerst naar Hendrik te gaan voor deze vraag. Hij wist daarom dat Hendrik zijn probleem wel kon oplossen. Als van Emde Boas niet een vriend was geweest, zou hij waarschijnlijk naar Amsterdam zijn gegaan om zijn probleem op te lossen. Van Emde Boas en Marchetti waren toch vrij theoretisch ingesteld, maar hadden nog nooit echt goed meegekregen hoe ze hun probleem goed konden oplossen. Terwijl Hendrik eigenlijk vrijwel meteen een oplossing uit zijn mouw schudde.

Het viel ons op dat de drie L's eigenlijk helemaal niet verbaasd waren dat er veel toepassingen zouden zijn. Hendrik was zelf niet met deze toepassingen bezig. Hij heeft een stelling en geeft hem door. Zoals Hendrik dat mooi verwoordde: "Na één stap voel ik dat ik al niet meer thuis ben." Hendrik hield zich dus niet met de toepassing bezig, net als Arjen.

László Lovász was net als Hendrik Lenstra zeer zuiver gemotiveerd. Hij is een combinatoricus en was destijds bezig met combinatorische optimalisatie. Lovász was bezig met de ellisoïdemethode. Hij kwam er achter dat de basisreductie van roosters, die Hendrik Lenstra had bedacht, hiervoor zeer nuttig was. Hij wou deze methode dus optimaliseren. Wat ideaal zou zijn, is dat dit algoritme uiteindelijk polynomiaal wordt. Want "een P algoritme is een 'goed' algoritme," volgens Lenstra. In die tijd was dit nog theoretisch, maar je had er wel echt iets aan als een algoritme aantoonbaar P was.

Hendrik Lenstra had geen 'verkooppraatje' voor zijn onderzoek. Dit was puur omdat zijn 'doel' niet lag bij de toepassingen. Of Universiteit Leiden soms contractresearch uitvoert wist hij niet precies – hij gaat uit van wel – maar zijn afdeling doet dat niet. Hij zegt dat geld bij het bedrijfsleven ophalen voor de wiskunde sowieso heel lastig is: "De eerste en tweede geldstroom, daar hield en houdt het eigenlijk wel mee op."

Ons leek de correspondentie tussen Lovász en Lenstra uit het niets te komen. Daarom vroegen we hoe wiskundigen corresponderen en correspondeerden. Het blijkt dat je niet zozeer direct naar een specialist gaat, maar eerst gaat aankloppen bij mensen van je directe omgeving. Hierbij gaat het bijvoorbeeld over vrienden en familie. "Je gaat niet direct naar een wildvreemde."

Het algoritme heeft nog wel wat veranderingen ondergaan na de publicatie van het artikel. Maar dit waren verandering over de snelheid van het algoritme. Hierbij gaat het vaak over het begrenzen van tussenresultaten. Hendrik Lenstra zag zijn rol in het bewijzen dat het algoritme polynomiaal is, met een zo elegant en kort mogelijk bewijs. "Ik heb de grootste stap gedaan. Ik haalde het van oneindig naar twintig. De rest mag het van twintig naar twee brengen."

Ten slotte vroegen we Hendrik Lenstra waar hij de ontwikkeling van het algoritme zou plaatsen in de modellen van Stokes en Bush, die we kort uiteen zetten. Ook vroegen we hem hoe hij in het algemeen denkt over de vragen die deze modellen proberen te beantwoorden. Hij plaatste zichzelf in het kwadrant van Bohr. Hij noemde wel enkele namen van wetenschappers die in het kwadrant van Pasteur zouden passen, waaronder die van Alfio

Quarteroni. Desalniettemin is hij blij dat zijn theorieën toegepast worden in de praktijk. “Ook probeer ik in mijn colleges altijd te laten zien aan studenten hoe de theorie toepasbaar is voor het oplossen van een aansprekend concreet probleem.”

A.2 Ionica Smeets

Ionica Smeets kwam in aanraking met het LLL-algoritme door haar promotieonderzoek. Haar onderzoek ging over kettingbreuken in meerdere dimensies. Het bleek dat het LLL-algoritme nuttig was voor het benaderen van deze kettingbreuken. Omdat zij actief is en was als wetenschapsjournaliste, was het idee van Hendrik Lenstra dat zij hierover een artikel schreef. Een extra motivatie hiervoor was dat ze het een bijzondere ontstaansgeschiedenis vond.

Het artikel werd uiteindelijk niet gepubliceerd in een wetenschappelijk tijdschrift, maar wel nog in een jubileumboek over het LLL-algoritme. Maar van de ene kant is het te begrijpen dat dit soort abstracte dingen niet interessant zijn voor een algemeen publiek. Toepassingen in bijvoorbeeld JPEG zijn daarentegen veel interessanter. Vaak zijn dingen die wiskundigen interessant vinden te ‘moeilijk’ voor een algemeen publiek. Toch hoeft het niet allemaal zo moeilijk te zijn. Een goed voorbeeld hiervan zijn de Wiskundemeisjes, waarvan Ionica er één is, die onder meer een column in *de Volkskrant* hebben. Zij ervaart dat je een eenvoudig maar belangrijk wiskundig principe goed begrijpelijk kunt maken.

Wat wel opviel was dat Hendrik Lenstra, Arjen Lenstra en László Lovász het artikel over het LLL-algoritme bewust op een plek terecht kwam waar ‘echte’ wiskundigen het zouden lezen en niet alleen informatici. De reden was dat ze graag hadden dat ook deze wiskundigen in aanmerking zouden komen met complexiteit.

Verder bevestigde Smeets de geschiedenis van het LLL-algoritme. Hieruit volgde dat Lovász destijds veel problemen had met het reizen buiten Hongarije. Dit kwam door het het IJzeren Gordijn: ’maximale kennis binnenhouden en niet te veel afgeven’ was het Sovjetdenken over wetenschap. Lovász had de meeste invloed op de ontwikkeling van het LLL-algoritme als zodanig: “Ze hadden het net zo goed het Lovászalgoritme kunnen noemen.”

Nadat we haar hadden verteld over het model van Stokes, zei ze dat ze het LLL-algoritme in het kwadrant van Pasteur vond passen. Maar dat hangt er natuurlijk sterk van af wat je precies als ‘toepassing’ ziet: bedoel

je uitsluitend praktische toepassingen of ook toepassingen in concrete maar voornamelijk wiskundig interessante vraagstukken?