

**Rob Bisseling**

Corresponderend auteur 'Software partitionering' *Mathematisch Instituut Universiteit Utrecht*  
 Postbus 80010  
 3508 TA Utrecht  
 rob.bisseling@math.uu.nl

**Alistair Vardy**

Corresponderend auteur 'Spieren' *Faculteit der Bewegingswetenschappen Vrije Universiteit*  
 Van der Boechorststraat 9  
 1081 BT Amsterdam  
 a.vardy@fbw.vu.nl

**Mark Peletier, Natacha Severens**

Corresponderende auteurs 'Hart-longmachine' *Faculteit Wiskunde en Informatica Technische Universiteit Eindhoven*  
 Postbus 513  
 5600 MB Eindhoven  
 m.a.peletier@tue.nl

**Studiegroep wiskunde met de industrie**

# Brainstormen voor bruikbare wiskunde

Op de studiegroep Wiskunde met de Industrie in 2005 in Amsterdam was de medische wetenschap prominent aanwezig. Het woord 'industrie' in de naam van de workshop moet dan ook al sinds vele jaren ruim worden opgevat. De studiegroep is daarmee tevens een test hoe bruikbaar wiskunde eigenlijk is voor de samenleving. Hoe kunnen wiskundige disciplines commercieel worden ingezet, wat hebben we aan de nieuwste ontwikkelingen in de statistiek, wat is de betekenis van de steeds maar groeiende kennis van het modelleren met differentiaalvergelijkingen? Soms valt dit tegen, maar wiskundige common sense geeft de bedrijven ook dikwijls een andere blik op hun probleem. En de wiskundigen kunnen met hun neus bovenop een open-hartoperatie komen te staan.

Bij de bedrijven die de zes problemen van de studiegroep hadden aangeleverd, zaten twee bekenden: het *Forensisch Instituut* en de *KLM*. Nieuw waren het *Nederlands Meetinstituut*, *Software Improvement Group*, het *Institute for Fundamental and Clinical Movement Sciences* en het *Academisch Medisch Centrum*. De onderzoeksresultaten van de laatstgenoemde drie bedrijven worden in dit artikel besproken, de eerste drie komen in het volgend nummer aan bod.

**Wiskundige neuromusculair onderzoek (Alistair Vardy)**

Binnen de neurowetenschappen zijn er tegenwoordig ontzettend veel manieren om onderzoek te doen. Ethische en praktische factoren beperken echter de toepasbaarheid van veel technieken bij onderzoek naar het zenuwstelsel. In veel spieronderzoek is het niet wenselijk en soms zelf vervelend voor de proefpersoon om direct aan het zenuwstelsel zelf metingen te verrichten. Metingen aan de buitenkant van het lichaam geven een ratjetoe van alle informatie die zich onder de huid afspeelt. Het ontrafelen van deze informatie is waar we in dit artikel op uit zijn.

*Alpha-motorneuronen en motoreenheden*

We gebruiken onze spieren om te bewegen. Ons zenuwstelsel activeert onze spieren door er elektrische pulsen naartoe te sturen. Deze elektrische pulsen gaan over het spieroppervlak waardoor de spier samentrekt. Als we beter kijken, zien we dat spieren zijn opgebouwd uit kleinere eenheden, spiervezels genaamd. Eén spier wordt aangestuurd door een heleboel zenuwen. Deze zenuwen heten alpha-motorneuronen. Elk van deze zenuwen stuurt een aantal spiervezels aan. Een alpha-motorneuron en zijn spiervezels vormen een geheel, dat we een motoreenheid zullen noemen. Kleine spieren hebben 10 van deze motoreenheden. In grote spieren kan dit aantal oplopen tot 300.

Als we onze spieren aanspannen, zijn er motoreenheden actief. Niet alle motoreenheden zijn tegelijk actief. Als we onze spieren vrijwillig aanspannen worden eerst de kleinere motoreenheden geactiveerd en pas bij zwaardere inspanning ook de grotere. Bij vrijwillige inspanning is de recruteringsvolgorde dus van klein naar groot. Op deze manier kunnen we op verschillende inspanningsniveaus de kracht goed doseren.

Naast vrijwillige inspanning is het ook mogelijk een spier kunstmatig te activeren. Door kleine elektrische schokjes toe te dienen, zal de spier samentrekken. Als er op deze manier inspanning wordt geleverd, dan is de recruteringsvolgorde van groot naar klein. Bovendien worden bij kunstmatige stimulatie van een spier alle motoreenheden op hetzelfde moment geactiveerd, terwijl bij vrijwillige inspanning de motoreenheden op verschillende tijdstippen worden geactiveerd.

Bij spierziekten zoals polio of Duchennespierdystrofie verzwakken de spieren. Bij polio komt dit doordat de verbindingen tussen de alpha-motorneuronen en de spiervezels afsterven. Als een verbinding afsterft, dan raakt de spier een motoreenheid kwijt. Bij Duchennespierdystrofie zorgt het ontbreken van een

specifiek spiereiwit ervoor dat spiercellen afsterven. Ook dit leidt tot een vermindering van motoreenheden. Om de progressie van spierziekten bij te kunnen houden, zouden we graag weten hoeveel motoreenheden een spier bevat.

In dit artikel bekijken wij een techniek om het aantal motoreenheden in een spier te bepalen. Een veelgebruikte techniek bij neurologisch onderzoek naar spieren is het meten van de elektrische activiteit op de huid. Een probleem hierbij is dat de gemeten signalen niet alleen van de spieren komen. Elektrische signalen van het hart en de andere organen worden ook door de huid geleid. De geleidende eigenschappen van de huid worden ook benut door bijvoorbeeld leugendetectors. Het totaal dat we meten, is een combinatie en vervorming van signalen. Gelukkig is de activiteit van spieren krachtig. Als we vlak boven een spier meten, is dit signaal krachtig genoeg om de overige signalen te overstemmen. Het meten van de elektrische activiteit van de spieren wordt elektromyografie (EMG) genoemd.

Onze interesse gaat uit naar het meten van de activiteit van de motoreenheden. Om nauwkeurige metingen te verrichten, worden de spieren bedekt met een matje van elektroden. Bij een vrijwillige of (met elektrische schokjes) gestimuleerde inspanning zien we activiteit die is opgebouwd uit de activiteit van verschillende motoreenheden. Helaas liggen de spiervezels van motoreenheden niet netjes gebundeld, maar kriskras door elkaar. Dit betekent dat de elektrische activiteit die binnenkomt bij elke elektrode op het matje een mengelmoes van signalen is.

Om het probleem te illustreren, voeren we een gedachtenexperiment uit. Stel we zouden elk van de motoreenheden één voor één kunnen activeren. Elke motoreenheid zou dan zijn eigen 'vingerafdruk' achterlaten. Als we elke vingerafdruk zouden kennen, dan is het probleem gereduceerd tot het bepalen van het aantal verschillende vingerafdrucken in de gemeten signalen. Helaas geven de motoreenheden hun vingerafdrucken niet zomaar prijs. Een manier om dit probleem te omzeilen, is om een serie van elektrische schokjes toe te dienen in oplopende sterkte. Bij de eerste schokjes is slechts de grootste motoreenheid actief. Daarna komen één voor één de kleinere motoreenheden erbij. We zien op deze manier de optelsom van de vingerafdrucken. Door van het gemengde signaal de vorige af te trekken, krijgen we de individuele vingerafdrucken. Deze aanpak werkt mits het totaal aan signalen niet te vervuild is. De vervuiling die optreedt, noemen we ruis. Het is noodzakelijk dat de activiteit van de motoreenheden groter is dan de ruis om ze nog te kunnen onderscheiden. Op dit moment wordt dit proces met de hand uitgevoerd. Dit is echter een tijdrovende bezigheid die moet worden uitgevoerd door een expert, en waarvoor geavanceerde apparatuur nodig is. Ons doel is een manier te vinden om het aantal motoreenheden te bepalen op een automatische, eenvoudig uit te voeren wijze. Als we aannemen dat alle vingerafdrucken uniek zijn, hoeven we alleen te weten hoeveel verschillende componenten er aanwezig zijn in het totaal van signalen. Hierbij biedt wiskunde een uitkomst.

#### Bepaling van principale componenten voor data met vertragingen

Gegeven een verzameling data  $X \in \mathbf{R}^{m,n}$  is het een klassieke taak om het aantal principale componenten te bepalen. Over de gemeten data wordt typisch aangenomen dat ze van de vorm

$$X_{ij} = \sum_{k=1}^N C_{ik} v_k(t_j) + \eta_{ij}$$

zijn, waar  $X, \eta \in \mathbf{R}^{m,n}$ ,  $C \in \mathbf{R}^{m,N}$ , en  $\{t_j\}_{j=1}^m$  zijn de tijdstippen waarop het signaal gesampled is. Dat wil dus zeggen dat de  $n$  signalen kunnen worden uitgedrukt als lineaire combinaties van een klein aantal  $N$  ( $N \ll n$ ) basisvectoren, plus ruis  $\eta_{ij}$ . In dit probleem is het voornamelijk van belang om de dimensie  $N$  van de opspannende verzameling  $\{v_k\}$  te bepalen, corresponderend met het aantal actieve motoreenheden.

Bij afwezigheid van ruis kan het probleem worden opgelost door de data te ontbinden in hun *principale componenten* door middel van een *singuliere-waardendecompositie*: we vinden matrices  $U \in \mathbf{R}^{m,m}$ ,  $V \in \mathbf{R}^{n,n}$ ,  $\sigma \in \mathbf{R}^m$  zodat

$$U^T X V = [\text{diag}(\sigma) \ 0_{n,m-n}].$$

Bovendien zijn  $U$  en  $V$  orthogonale matrices en  $\sigma_i \geq \sigma_{i+1} \geq 0$ . Van belang hierbij is dat als de matrix rang  $N < n$  heeft, dan  $\sigma_{N+1} = \dots = \sigma_n = 0$ . In dat geval zijn er precies  $N$  principale componenten. In het algemeen zijn echte meetgegevens niet zo netjes, en experimentele afwijkingen zorgen ervoor dat  $\sigma_i > 0$  voor alle  $i$ . Maar de relatieve grootte van de principale waarden  $\sigma_i$  kan ons nog steeds veel informatie geven. In het bijzonder is  $\sigma_{k+1}$  gelijk aan de afstand van  $X$  tot de verzameling van matrices met rang  $k$ :

$$\sigma_{k+1} = \min_{\text{rang}(Y)=k} \|X - Y\|_2.$$

Gegeven een drempelwaarde  $\varepsilon$  kunnen we de  $\varepsilon$ -rang  $r_\varepsilon$  van een matrix te definiëren door te eisen dat

$$\sigma_{r_\varepsilon} > \varepsilon \geq \sigma_{r_\varepsilon+1}.$$

In het voorliggende probleem kunnen we deze ideeën niet direct toepassen omdat er *twee* verschillende bronnen van fouten zijn:

1. Experimentele fouten van onbekend type en grootte *in* elk kanaal ('ruis').
2. Vertragingen van onbekende grootte *tussen* alle paren van signalen.

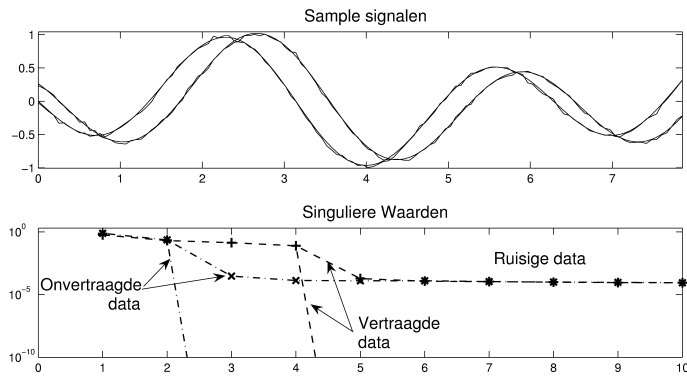
Fouten van de eerste soort zijn wijdverbreid en niet bijzonder problematisch; fouten van het tweede type moeten we nader bekijken. De metingen beginnen niet allemaal op hetzelfde tijdstip in relatie tot het begin van het signaal in de spieren. Hierdoor zijn de metingen ten opzichte van elkaar enigszins vertraagd. Laten we aannemen dat het tijdvertraagde signaal gegeven wordt door (cf. (1))

$$\hat{X}_{ij} = \sum_{k=1}^N C_{ik} v_k(t_j + s_i) + \eta_{ij},$$

waarbij  $s_i$  de tijdsvertraging in de  $i$ -de meting is. Dan

$$\begin{aligned} \hat{X}_{ij} &= \sum_{k=1}^N C_{ik} \left( v_k(t_j) + s_i v'_k(t_j) + \frac{s_i^2}{2} v''_k(t_j) + \dots \right) + \eta_{ij} \\ &\simeq \sum_{k=1}^N C_{ik} \left( v_k(t_j) + s_i v'_k(t_j) \right) + \eta_{ij} \quad (\text{bij kleine vertragingen}) \\ &= \sum_{k=1}^N C_{ik} v_k(t_j) + \sum_{k=1}^N \tilde{C}_{ik} \tilde{v}_k(t_j) + \eta_{ij}. \end{aligned}$$

De functies  $\tilde{v}_k$  kunnen nu niet als lineaire combinaties van de  $v_k$  geschreven worden, dus zelfs zonder ruis zouden we de rang van de matrix en de ogenschijnlijke dimensie van de basis van principale componenten hebben verdubbeld.



**Figuur 1** Boven: voorbeeldsignaal met typische vertraging en kleine ruis. Onder: de singuliere waarden voor de data zonder ruis nemen af tot  $\sigma_i = 10^{-40}$  na  $N = 2$  respectievelijk  $N = 4$  voor de onvertraagde en vertraagde data. Kiezen we in dit voorbeeld  $\varepsilon = 10^{-3}$  dan doet de toevoeging van ruis niets af aan de bepaling van de  $\varepsilon$ -rang, maar de vertragingen verdubbelen de schatting voor het aantal principale componenten.

We geven een klein voorbeeld om dit te illustreren. Beschouw matrices van de vorm

$$X_{ij} = \alpha_i \sin(t_j) + \beta_i \sin(2t_j) \quad , i = 1, \dots, 10, t_j \in [0, 5\pi]$$

$$\hat{X}_{ij} = \alpha_i \sin(t_j + s_i) + \beta_i \sin(2t_j + s_i)$$

waar  $\alpha_i, \beta_i, s_i$  willekeurig gekozen zijn in  $[-1, 1]$ , en  $\varepsilon = 10^{-3}$ . In dit voorbeeld hebben we  $N = 2$  en  $n = 10$  genomen. In figuur 1 tonen we een signaal met en zonder vertragingen en ruis en daarvoor de bijbehorende singuliere waarden.

Dit voorbeeld toont dat we de singuliere waarde decompositie niet zomaar kunnen toepassen, zelfs niet met een drempelwaarde, om een goede schatting te krijgen voor de dimensie van de basis van principale componenten voor de vertraagde data met ruis. Om het aantal principale componenten te bepalen gaan we in twee stappen te werk. Eerst nemen we aan dat we de dimensie van de basis weten en zoeken we de beste/juiste verschuivingen, die de tijdsvertragingen ‘opheffen’. Daarna zullen we proberen de juiste dimensie te bepalen.

Het komt er dus op neer dat we de verschuivingen zo zouden willen kiezen dat ze  $\sigma_{N+1}$  minimaliseren, even aannemende dat we  $N$  weten. In de praktijk blijkt dat echter een zeer gedegeneerd en moeilijk probleem. In plaats daarvan bekijken we

$$V = - \frac{\sum_{i=1}^N \sigma_i}{\sum_{i=1}^n \sigma_i} = - \sum_{i=1}^N \sigma_i \quad (\text{door normalisatie}).$$

Hierbij nemen we dus aan dat  $N$  een vast getal is. Als de singuliere waarden snel afnemen voor  $i > N$  dan

$$- \sum_{i=1}^N \sigma_i \simeq - \sum_{i=1}^n \sigma_i + \sigma_{N+1} \simeq -1 + \sigma_{N+1},$$

en in zulke gevallen is de minimalisatie van (1) een goede benadering voor  $\sigma_{N+1}$ .

Het is goed om hier op te merken dat singuliere waarde decompositie een *linear* probleem is, terwijl het vinden van de beste verschuivingen (om de vertragingen te compenseren) een essentieel *niet-linear* probleem is. We hebben drie methoden getest om de functionaal in (1) te minimaliseren met betrekking tot de verschuivingen:

- A1. Een pseudo-Newton methode voor het vinden van een nulpunt van  $\nabla V$  met behulp van minunc in de *Optimalisatie Toolbox* van Matlab.
- A2. Een gradiëntevoortje van  $V$  met betrekking tot een artificiële tijd door middel van

$$\frac{ds_i}{dt} = - \frac{\partial V}{\partial s_i}$$

via de code ode113 in Matlab.

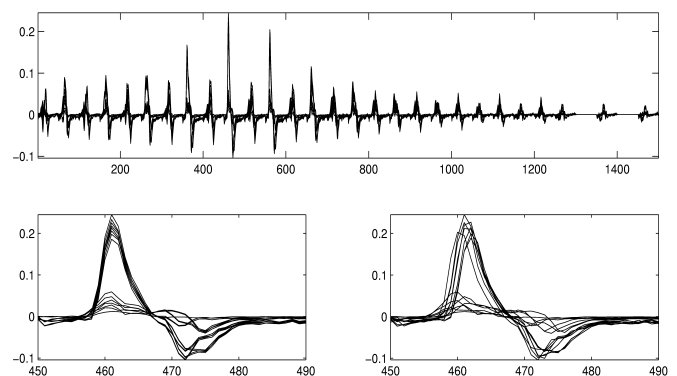
- A3. Direct zoeken met de Matlab routine `fminsearch`. De resultaten van de algoritmes A1 en A2 zijn vrijwel identiek, maar A2 kost ongeveer tien keer zoveel tijd. De resultaten van A3 zijn minder bevredigend en zijn tien keer zo langzaam als A2.

We nemen nu aan dat we redelijke schattingen hebben voor het minimale aantal  $N_{\min}$  en het maximale aantal  $N_{\max}$  principale componenten. Dan voeren we de minimalisatie procedure uit met betrekking tot de verschuivingen. Dit leidt, voor vaste  $N$ , tot de beste teruggeschoven data.

In het ideale geval zouden we graag een duidelijk stap willen zien van grote naar kleine singuliere waarden (zoals in het voorbeeld in figuur 1), zodat we een duidelijke  $\varepsilon$ -rang kunnen bepalen. Helaas lijkt het niet mogelijk om op goede gronden een waarde van  $\varepsilon$  te kiezen. In plaats daarvan bepalen we voor elke (vaste)  $N \in [N_{\min}, N_{\max}]$  de maximale verhouding tussen opeenvolgende singuliere waarden:

$$m(N) = \max_k \frac{\sigma_k}{\sigma_{k+1}}.$$

Laat  $N_r(N)$  de waarde van  $k$  zijn waarvoor dit maximum wordt aangenomen.  $N_r(N)$  is dan de schatting voor het aantal principale componenten als we de verschuivingen geoptimaliseerd hebben voor de  $N$  grootste componenten. Voor consistentie is duidelijk vereist dat voor het werkelijke aantal principale componenten geldt dat  $N_r(N) = N$ . Natuurlijk kunnen er meerdere waarden van  $N$  zijn waarvoor aan deze consistentie-eis is voldaan. Als beste schatting voor het werkelijk aantal principale componenten, kiezen we de  $N$  die de verhouding  $m(N)$  maximaliseert. Conceptueel vindt deze benadering de dimensie waarin er de duidelijkste stap van ‘grote’ naar ‘kleine’ componenten is. Aan de andere kant, dit sluit niet uit dat de data in elke meting bestaat uit één hele grote component en verscheidene kleinere. In dat geval blijkt



**Figuur 2** Artificiële data; boven: het complete signaal in alle kanalen; linksonder: detail van de onvertraagde data; rechtsonder: detail van de vertraagde data. Na optimalisatie van de verschuivingen zijn de originele data en de teruggeschoven data in essentie niet te onderscheiden.

het erg moeilijk om de kleine signalen van de ruis te onderscheiden omdat de vertragingen het klassieke signaal-versus-ruis probleem versterken. Samenvattend is het totale algoritme als volgt:

1. Minimaliseer  $V$  in (1) met betrekking tot verschuivingen,  $N \in [N_{\min}, N_{\max}]$ .
2. Bepaal  $m(N)$  in (2) voor de teruggeschoven data.
3. Maximalisatie van  $m(N)$  met betrekking tot  $N$  leidt tot de schatting van het aantal principale componenten.

In figuur 2 laten we een test met artificiële data zien. Deze zijn ‘gesimuleerd’ door een van de experimentatoren. De resultaten van het algoritme zijn samengevat in tabel 1.

$N$	$N_r$	$\sigma_{N_r} / \sigma_{N_r+1}$	Rekentijd (s)
6	6	125	47
5	5	98	45
4	4	2731	33
3	4	113	39
2	4	31	31

**Tabel 1** Resultaten van het algoritme met verschaft ‘gesimuleerde’ data. We concluderen hieruit terecht dat er vier principale componenten zijn.

## Conclusies

We hebben gezien dat het algoritme succesvol is in het reduceren van het effect van de vertragingen in de data, zodat het een betrouwbare schatting voor het aantal principale componenten geeft. Na voltooiing van dit verslag zijn we voorzien van ‘echte’ data, en onze voorlopige conclusie is dat de benadering met het beschreven algoritme hierbij ook goed lijkt te werken, maar dat de gradiënt evolutie (algoritme A2) het meest geschikt is. Hier moet nog diepgaander naar gekeken worden. Er is nog een punt waarop progressie kan worden geboekt. We hebben namelijk geen informatie gebruikt over de structuur van de signalen, terwijl dat ons toch zou moeten kunnen helpen bij het ontrafelen van het juiste aantal principale componenten.

De vraag die we hier bekeken hebben bestaat eigenlijk uit twee componenten. Hoe kunnen we de data terugschuiven zodat de vertragingen worden opgeheven? En hoe kunnen we daarna het aantal principale componenten bepalen? We denken dat we de eerste vraag beter kunnen beantwoorden. Dat is niet verrassend, want de tweede vraag is een oud probleem dat niet volledig is opgelost in veel praktische situaties.

Een andere manier om inzicht te krijgen in het werken van motoreenheden is te kijken naar het gedrag van de zenuwen die de spieren aansturen, de alpha-motorneuronen. In ditzelfde onderzoek hebben we ook een stochastisch model voor een alpha-motorneuron opgesteld om het activatiegedrag te onderzoeken. Een zenuw krijgt invoer van andere zenuwen in de vorm van elektrische pulsjes. De zenuw reageert op het totaal van deze pulsen door zelf ook pulsen te genereren. Een alpha-motorneuron krijgt zijn invoer van het centrale zenuwstelsel en stuurt zijn pulsen naar zijn spiervezels. Het model is ontworpen om pulsen van het centrale zenuwstelsel op te vangen en daarop zelf pulsen te sturen. Simulaties van het model geven een realistisch beeld van het pulsgedrag van een alpha-motorneuron. ◀

Wiskundigen die aan het neuromusculair onderzoek werkten: JF Williams, Geertje Hek, Alistair Vardy, Vivi Rottschäfer, Jan Bouwe van den Berg en Joost Hulshof.

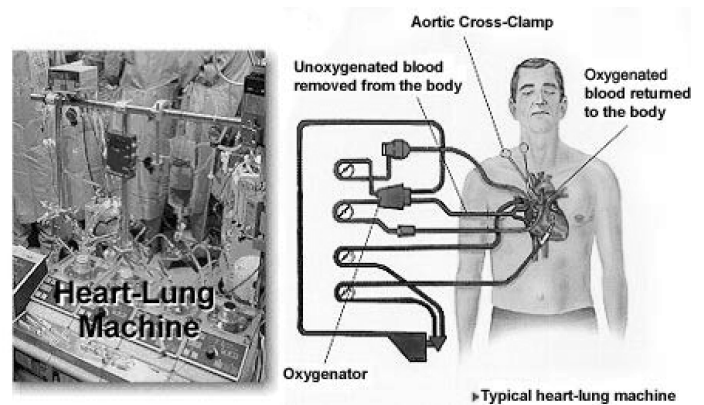
## Modelvorming van het koel- en opwarmproces van patiënten tijdens openhartoperaties (Natacha Severens en Mark Peletier)

Voor hen die het leven te kort vinden is *cryogene suspensie* — conservering van het lichaam door bevroering — altijd een fascinerende gedachte geweest. Voor de meeste wetenschappers en medici is het ook niet meer dan dat: tot nu toe is suspensie en opwarming alleen geslaagd bij de simpelste organismen, zoals nematode wormen. Dit weerhoudt sommige optimisten er niet van om tegen torenhoge kosten afspraken te maken met organisaties als de Alcor Life Extension Foundation, in de hoop over duizenden jaren te ontwaken in een betere wereld — of op z'n minst in een wereld waar de ouderdom succesvol verlengd kan worden.

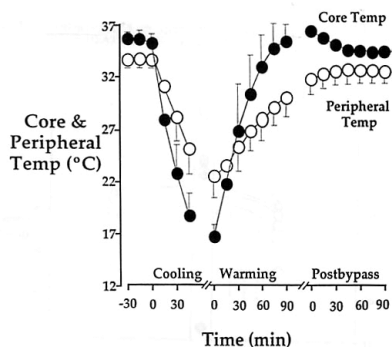
Weinigen beseffen dat afkoeling en opwarming routine is in hedendaagse ziekenhuizen. Tijdens *cardiopulmonaire bypass* operatie, een type operatie waarbij de functie van het hart en de longen tijdelijk door een *hart-longmachine* wordt overgenomen, wordt de patient vaak vijf of tien graden gekoeld, en soms zelfs tot twintig graden toe — tot een temperatuur van 17 graden Celsius. Vergelijk dit met de ervaring dat onderkoelde bergbeklimmers bij een kerntemperatuur van 25 graden al nauwelijks te reanimeren zijn. Zonder *life-support system* was bij dergelijke onderkoeling de patient al dood geweest.

Een deel van de afkoeling tijdens zo'n open-hartoperatie is niet eens opzettelijk. De patient verliest alleen al warmte door de vaatverwijding die de toegediende anesthetica met zich meebrengen, door de koude operatiekamer, door de opening van de borstkas, waardoor een groot, nat oppervlak vrijkomt, en doordat het bloed in contact komt met de koude hart-longmachine. De hart-longmachine bevat daarom een warmtewisselaar waarmee de perfusionist — de specialist verantwoordelijk voor de buitenlichamelijke bloedsomloop — de temperatuur van het bloed kan regelen.

Als de temperatuur van het bloed toch geregeld kan worden, waarom wordt de patient dan zo ver afgekoeld? Dat heeft te maken met risicobeperking. Bij een open-hartoperatie leidt een onverwachte situatie (een gesprongen slagader, bijvoorbeeld) al snel tot een onderbreking van de bloedsomloop. De hersenen zijn de grootste zuurstofverbruiker van het lichaam, en bij gebrek aan zuurstof ontstaat snel hersenbeschadiging. De zuurstofbehoefte van cellen neemt echter sterk af als de temperatuur daalt. Door het lichaam te koelen heeft de chirurg veel meer tijd om een mogelijke crisis te bezweren.



Hart-longmachine



Temperatuurverloop tijdens een operatie (dichte bollen: kern, open bollen: periferie) [1]

Zolang de patient aan de hart-longmachine verbonden is gaat dit proces goed. De vitale organen, vooral de hersenen, zijn zeer goed doorbloed, en nemen de temperatuur van het bloed snel over. De problemen ontstaan pas wanneer de hart-longmachine wordt afgekoppeld, en de patient weer zijn eigen temperatuur moet gaan regelen. Dit leidt altijd tot een snelle daling van de kerntemperatuur met soms wel enkele graden, de zogeheten *afterdrop*.

Die *afterdrop* is erg ongelukkig, vooral bij een hartoperatie. De patient krijgt het koud en begint te rillen, waardoor veel energie en zuurstof verbruikt wordt, en dat betekent een grotere belasting voor het hart. Hoe kleiner de *afterdrop*, hoe beter het genezingsproces, zo is de ervaring, en de centrale vraag aan de Studiegroep was om deze *afterdrop* beter te begrijpen.

De oorsprong van de *afterdrop* ligt in het verschil in doorbloeding tussen de 'kern' van het lichaam, de romp en de hersenen, en de 'periferie', bestaande uit de armen en benen. Tijdens de urenlange operatie koelt de periferie af; na opwarming door de hart-longmachine is de temperatuur van de goed doorbloede kern wel weer op niveau, maar die van de minder goed doorbloede periferie loopt een paar graden achter. Na afkoppeling wordt de warmte herverdeeld: de kern warmt de periferie op, en wordt daardoor zelf kouder.

### Modelvorming

Aan de basis van modelvorming staan de transportvergelijkingen die de warmtewisselingsprocessen beschrijven die plaatsvinden binnen het lichaam van de patient en tussen patient en omgeving. Tijdens de narcose spelen actieve regelmechanismen zoals zweten, rillen en vasoconstrictie geen rol; het lichaam gedraagt zich, wat warmte betreft, als passieve materie. Weefsels veranderen dan op twee manieren van temperatuur: door warmteuitwisseling met het bloed (perfusie) of door warmtegeleiding met naburige weefsels.

Waar alle weefsels warmte ongeveer even goed geleiden — we bestaan tenslotte voornamelijk uit water — zijn er grote verschillen in doorbloeding van verschillende typen weefsel. De 'kern' van het lichaam, de hersenen en de romp, is uitstekend doorbloed, zodat er nauwelijks verschil in temperatuur is tussen het hersenweefsel en het bloed dat na de warmtewisselaar het lichaam weer ingaat. Het 'perifeer' weefsel, voornamelijk de armen en benen, heeft een veel zwakkere bloedvoorziening, en is voor temperatuurveranderingen deels op geleiding aangewezen.

Het simpelste model voor de temperatuurveranderingen tijdens een operatie bevat dan twee compartimenten, kern en periferie. We hebben daar een derde compartiment aan toegevoegd;

het rectum, het gebied rondom de anus, is belangrijk omdat tijdens de operatie de chirurg beslissingen neemt over opwarming en afkoeling op basis van de rectale temperatuur.

### Resultaten

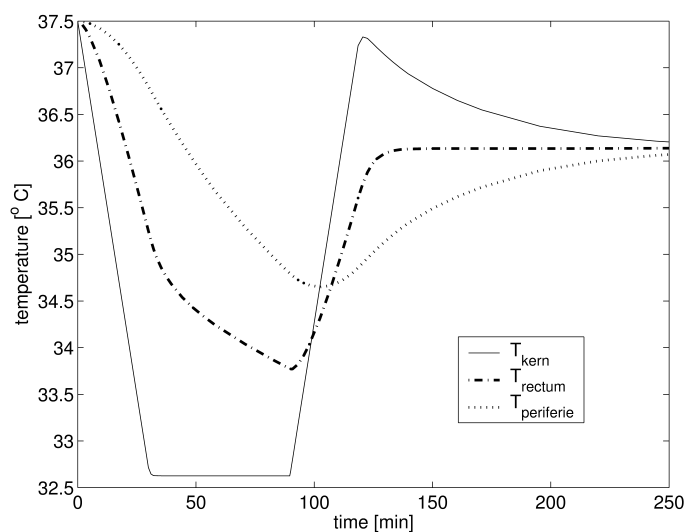
Met dit model is geprobeerd het verloop van experimenteel gemeten kern en rectale temperatuur te reproduceren. De operatie wordt opgesplitst in vier delen: afkoelen, constante (lage) temperatuur, opwarmen, en de periode na afkoppelen van de hart-longmachine; de vier delen zijn in de grafieken hieronder te herkennen. Dit model blijkt in staat te zijn om typische kenmerken van het temperatuurverloop tijdens een operatie te voorspellen, zoals de langzame reactie van de periferie ten opzichte van de kern en rectum en het optreden van *afterdrop*. Het blijkt dat de periferie zich in de afkoelfase als een warmtebron gedraagt en in de opwarmfase als een warmteput; deze warmteput is verantwoordelijk voor de *afterdrop*. Uit simulaties met het model blijkt dat de chirurg de *afterdrop* kan verkleinen door bijvoorbeeld

1. de periode van lage temperatuur zo kort mogelijk te maken (niet getoond), of
2. de patient aan het eind van de opwarmperiode wat langer aan de hart-longmachine gekoppeld te laten (zie grafieken in figuur 3).

### Conclusie

Wat is de medische stand met dit model opgeschoten? Sommige resultaten komen overeen met de ervaringen van de chirurg. Dat is wel eens frustrerend ('ja, maar dat wisten we allang!') maar het is een bevestiging van de voorspellende kwaliteit van het model, en betekent daarom dat er goede kansen zijn om met dit model verder te kunnen komen dan menselijke ervaring en intuïtie. Vooral wijzen de resultaten de weg naar uitgebreider onderzoek: een goed begrip van perifertemperatuur, bijvoorbeeld, en hoe die verandert met de tijd en met de kerntemperatuur, blijkt essentieel te zijn in het beheersen van *afterdrop*.

In een samenwerking tussen het AMC, de TU Eindhoven, en de Universiteit Maastricht is inmiddels een promotieproject gaan-



**Figuur 3** Snel afkoppelen betekent hoge *afterdrop*: als patient meteen na het opwarmen van de hart-longmachine aan zichzelf wordt overgelaten, is de eindtemperatuur (uiteinde-lijke waarde in de grafiek links, 35.5°C) hoger dan wanneer de patient een tijdje aangekoppeld blijft (rechts, uiteinde-lijke waarde 36.6°C)

de dat zich richt op precies dit verdere onderzoek. Hierin zullen allerlei effecten nauwkeuriger bestudeerd en beschreven worden, zoals de verdeling van de temperatuur over de verschillende ledematen, het verlies aan warmte door de open borstkas, en het effect van simpele maatregelen zoals kruiken en warme dekens over de benen. Wie weet blijkt na een uitgebreide wiskundige analyse dat grootmoeder altijd al het beste wist hoe je je voeten weer warm krijgt? ←

Aan het afkoelingsprobleem werkten de wiskundigen Joyce Aitchison, Christina Giannopapa, Vincent Guyonne, Miroslav Kramar, Simon van Mourik, Jasmina Panovska, en Marcus Tindall. Met dank aan Dirk-Jan Veldman en Bas de Mol van het AMC.

### Automatische partitionering van softwaresystemen (Rob Bisseling)

De afgelopen jaren heeft de vooruitgang in de informatietechnologie een enorme invloed gehad. Steeds meer organisaties, zoals banken en overheidsinstellingen, zijn afhankelijk van immer groeiende softwaresystemen om hun taken te kunnen uitvoeren. De grote complexiteit van deze systemen maakt het systeemonderhoud erg duur. Een mogelijke oplossing voor dit probleem bestaat in het opsplitsen van deze systemen in kleinere minder complexe modules. Iedere module wordt dan onderhouden door een kleine groep systeembeheerders.

Zo'n gepartitioneerd systeem heeft echter een *interface* nodig voor de communicatie tussen de modules. Dit betekent dat ieder programma dat door andere modules gebruikt wordt, ook wel een interfaceprogramma genoemd, voorzien wordt van extra communicatiefaciliteiten. De kosten voor het extra onderhoud van deze interface zijn evenredig met de grootte van de interface ofwel het totaal aantal interfaceprogramma's. Voor een goede partitionering is de omvang van de interface minimaal en zijn de modules voldoende klein. In de praktijk zijn voor het vinden van zo'n splitsing ervaren deskundigen nodig. Voor erg grote softwaresystemen is het echter bijzonder handig om gebruik te maken van automatische partitioneer algoritmen, die een goede suggestie kunnen geven. Het bedrijf Software Improvement Group (SIG, [www.sig.nl](http://www.sig.nl)) gevestigd in Diemen ontwikkelt dit soort hulpmiddelen.

#### Partitieprobleem van een call-graaf

Een *graaf* is een bekend wiskundig concept voor het representeren van een netwerk. Een graaf bestaat uit een aantal punten, *knopen* genaamd, die verbonden zijn door lijnen. Als de verbindingen ook een richting hebben, worden deze *pijlen* genoemd en spreken we van een *gerichte graaf*.

Een softwaresysteem kan worden gemodelleerd door een call-graaf, waarvan de knopen de programma's voorstellen. Als programma  $v$  programma  $w$  aanroept (een call) is dit equivalent met een pijl van knoop  $v$  naar knoop  $w$ . Een korte notatie hiervoor is  $v \rightarrow w$ . Een module kan nu worden gerepresenteerd door een deelverzameling van de knopen van de call-graaf. Dit betekent dat de omvang van een module gelijk is aan het aantal knopen van de bijbehorende deelverzameling. We noemen knoop  $v$  een *interfaceknoop* als hij een inkomende pijl heeft vanuit een knoop in een andere module. De grootte van de interface is nu gelijk aan het aantal interfaceknopen. Zodoende kan het vinden van een goede splitsing voor het softwaresysteem worden geformuleerd als het volgende partitioneringsprobleem van een call-graaf.

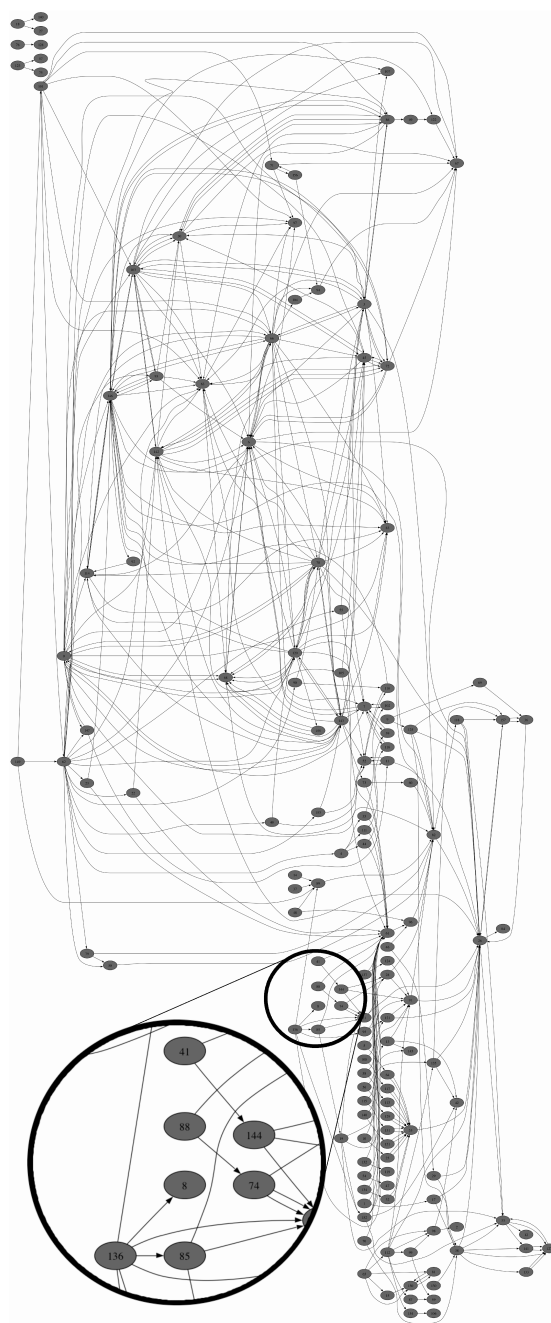
**Gegeven:** een call-graaf en de gehele getallen  $K, L$ .

**Vind:** een partitionering van de knopen van de call-graaf in  $L$  deelverzamelingen zodat iedere deelverzameling maximaal  $K$  knopen heeft en het totaal aantal interfaceknopen minimaal is.

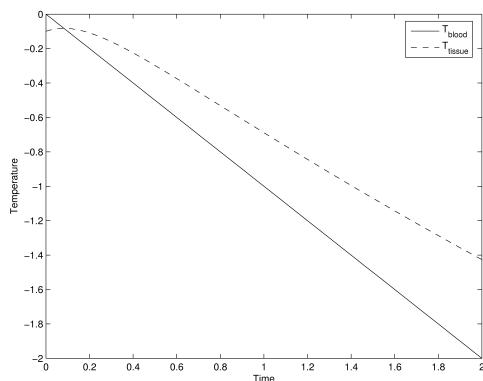
Figuur 4 laat een call-graaf zien van een softwaresysteem uit de praktijk met de naam Java1.

#### Verskillende oplosmethoden

Twee verschillende probleemformuleringen blijken veelbelovend voor het probleem van SIG. Allereerst modelleren we het probleem als een *geheeltallig lineair programmeringsprobleem* (in het Engels: *Integer Linear Programming, ILP*). We gebruiken daarbij be-



**Figuur 4** Call-graaf van het softwaresysteem Java1 met 158 knopen (programma's). De inzet rechtsonder geeft een gedetailleerde blik op een deel van de graaf waar de pijlen tussen de programma's duidelijk zichtbaar zijn.



Figuur 5 Verbindingsmatrix ( $158 \times 158$ ) van het probleem Java1.

slissingsvariabelen  $y_{vl}$  en  $x_{vl}$ , waarvan een waardering als volgt moet worden opgevat:

$$y_{vl} = \begin{cases} 1 & \text{knoop } v \text{ zit in deelverzameling } l \\ 0 & \text{anders} \end{cases}$$

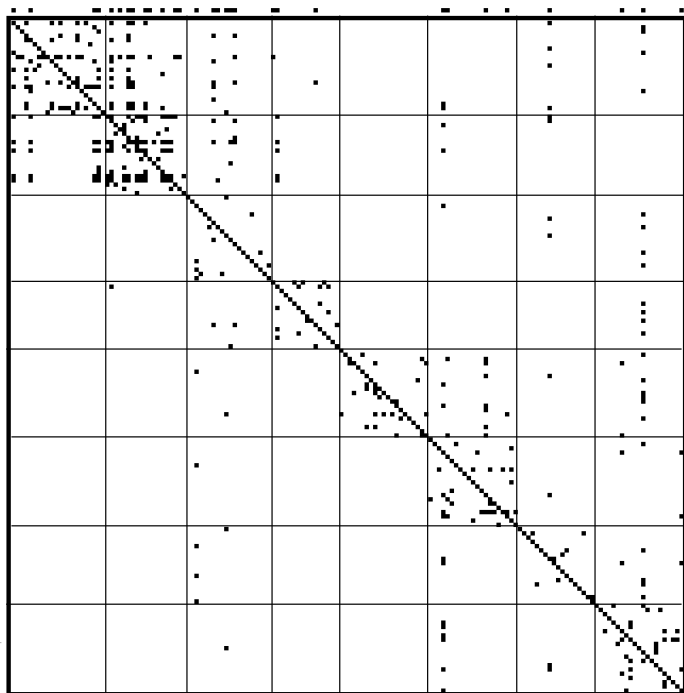
$$x_{vl} = \begin{cases} 1 & \text{knoop } v \text{ is een interfaceknoop} \\ & \text{bevat in deelverzameling } l \\ 0 & \text{anders.} \end{cases}$$

De eis dat module  $l$  niet meer dan  $K$  knopen mag bevatten kan dan geformuleerd worden als een lineaire ongelijkheid:

$$y_{v_1l} + y_{v_2l} + \dots + y_{v_Nl} \leq K.$$

En evenzo kan met lineaire vergelijkingen worden afgedwongen dat elke knoop in precies één module zit, etc. Het doel om zo min mogelijk interfaceknoten te gebruiken komt neer op het minimaliseren van de som van alle  $x_{vl}$ .

Deze formulering is een goede eerste stap naar een praktische oplossing. Een ILP probleem heeft een sterk meetkundig karakter,



Figuur 6 Gepartitioneerde verbindingsmatrix ( $158 \times 158$ ) van het probleem Java1. De rijen en kolommen zijn zodanig gepermuteerd dat ieder rij-blok bij één module hoort.

en er is een effectieve oplossingsmethode die de meetkundige aard van dit probleem uitbuit. Zodoende kan ook ons partitioneringsprobleem exact worden opgelost, maar de rekentijd loopt snel op als de grootte van het softwaresystemen toeneemt. Efficiënte herformuleringen en slimme hulpalgoritmen worden cruciaal om de oplossing te vinden binnen een redelijke tijd.

Als alternatief hebben we het probleem ook geformuleerd als een *hypergraafpartitioneringsprobleem*. Een hypergraaf is een generalisatie van een gewone ongerichte graaf met netten in plaats van lijnen. Een *net* verbindt een willekeurig aantal knopen, terwijl een lijn slechts twee knopen verbindt. Net  $n_j$  bestaat uit knoop  $j$  en alle knopen  $i$  met  $i \rightarrow j$ . Als de knopen in net  $n_j$  in verschillende deelverzamelingen vallen, is het net *gebroken*. Na toevoeging van de pijl  $j \rightarrow j$  voor alle knopen  $j$ , blijkt het aantal interfaceknoten gelijk te zijn aan het aantal gebroken netten. Het callgraafpartitioneringsprobleem kan dus ook worden geformuleerd als een hypergraafpartitioneringsprobleem (HP).

Een gerichte graaf met  $N$  knopen wordt gewoonlijk beschreven door haar verbindingsmatrix  $A = (a_{ij})_{i,j=1,\dots,N}$ , gedefiniëerd door

$$a_{ij} = \begin{cases} 1 & \text{als } i \rightarrow j, \\ 0 & \text{anders.} \end{cases}$$

Figuur 5 laat de verbindingsmatrix zien die hoort bij de callgraaf Java1. De verbindingsmatrix kan worden gepartitioneerd door zogenaamde multi-levelmethoden, zoals het matrixpartitioneringsprogramma Mondriaan. Dit programma werd aangepast, zodat het toegepast kon worden op de voorbeelden van SIG. Het is *heuristisch* van aard, maar desondanks bleek het erg snel en betrouwbaar zodat de resultaten dicht bij de optimale oplossing liggen.

#### Vergelijking van de twee algoritmen

We hebben twee methoden ontwikkeld voor het probleem, namelijk een optimaal ILP- en een heuristisch HP-algoritme. Het is een interessante vraag hoe de kwaliteit en rekentijd zich verhouden tussen deze twee varianten. Daarom zijn voor een aantal praktijkproblemen, geleverd door SIG, de beide methoden gebruikt om een optimale partitionering in  $L = 8$  modules te vinden, waarbij de maximale omvang van een module gelijk is aan  $K = 1.2 \frac{N}{L}$ . Het softwarepakket CPLEX is gebruikt op een 400 MHz Sun Ultra-Sparc II computer om de optimale oplossing te vinden, terwijl Mondriaan een 867 MHz Apple PowerBook G4 computer gebruikte. Figuur 6 laat het resultaat van het hypergraafpartitioneringsalgoritme zien voor het probleem Java1. Elke kolom correspondeert met een net; gebroken netten zijn bovenaan gemarkeerd.

In tabellen 2 en 3 is de interfaceomvang  $|I|$  en de rekentijd weergegeven voor beide methoden. De verschillende call-grafen hebben  $N$  knopen en  $|A|$  pijlen. Als we de resultaten vergelijken zien we dat de heuristische methode hele goede resultaten levert. De ILP methode leidt tot de optimale interface; de interface verkregen door de heuristische HP methode is groter, echter hoogstens een factor 1.63. Aan de andere kant is de rekentijd van de HP methode veel kleiner en lineair evenredig met  $|A|$ . De rekentijd van de ILP methode is over het algemeen veel hoger en heeft een onvoorspelbaar karakter.

Probleem	N	A	K	L	opt.  I	I /N (%)	rekentijd (s)
Java1	144	422	23	8	26	16.5	103
Java3	837	5252	127	8	242	28.4	246456 <sup>a</sup>
Java4	15	39	2	8	11	68.8	0.22
Cobol1	947	1900	209	8	13	0.9	118
Cobol2	449	659	81	8	6	1.1	351
Cobol3	1145	2686	203	8	51	3.8	6452
Cobol4	1100	2951	167	8	32	2.9	742172 <sup>a</sup>

**Tabel 2** Resultaten met 0-1 lineair programmeren. De met 'a' gemarkeerde tijden zijn afkomstig van een andere computer.

Probleem	N	A	beste  I	gem.  I	I /N (%)	rekentijd (s)
Java1	158	580	30	30.7	19.0	0.06
Java3	851	6103	275	283.2	32.3	0.54
Java4	16	55	11	11.2	68.8	0.001
Cobol1	1398	3298	17	22.4	1.2	0.33
Cobol2	545	1204	10	11.5	1.8	0.12
Cobol3	1357	4043	69	74.6	5.1	0.34
Cobol4	1116	4067	52	56.5	4.7	0.41

**Tabel 3** Resultaten met hypergraafpartitionering

### Conclusie

Het splitsen van een groot softwaresysteem in kleinere eenheden is voor veel organisaties een uitdaging geworden. Tijdens de studiegroep maakten we kennis met dit probleem, waar het werd geïntroduceerd door SIG. Tijdens de week van de studiegroep en erna hebben we gewerkt aan een duidelijke wiskundige formulering van het probleem. Tevens hebben we twee verschillende aanpakken gevonden, die beide erg geschikt en waardevol kunnen zijn voor de bedoelde toepassingen. De wiskundige formuleringen reduceerden het probleem tot standaardproblemen uit de discrete optimalisering. Zodoende konden moderne algoritmen gebruikt worden om de praktijkproblemen van SIG op te lossen. Onze twee methoden kunnen het relatieve aantal interfaceprogramma's drastisch verlagen. Voor Cobol-systemen is het zelfs mogelijk dat de omvang van de interface minder dan 5% wordt.

Voor kleine problemen bevelen wij aan de optimale ILP methode te gebruiken. Voor grote problemen, met duizenden knopen in de call-graaf, is multilevel hypergraafpartitionering de enige realistische optie.

Het bedrijf SIG waardeerde de voorgestelde oplossingen, evenals het inzicht dat verkregen kon worden door middel van ons onderzoek. Aan de andere kant hebben wij geprofiteerd van dit goed gestelde probleem wat ons leidde tot nieuwe interessante vragen zoals de vergelijking tussen een exacte en een heuristische methode in een realistische situatie.  $\Leftarrow$

Aan dit probleem werkten de wiskundigen Rob Bisseling, Jarosław Byrka, Selin Cerav-Erbas, Nebojša Gvozdenović, Mathias Lorenz, Rudi Pendavingh, Colin Reeves, Matthias Röger en Arie Verhoeven.

### Referenties

1 Tissue Heat Content and Distribution During and After Cardiopulmonary Bypass

at 17°C; Rajek, A., Lenhardt, R., Sessler, D.I., Grabenwöger, Kastner, J., Mares, P.,

Jantsch, U., and Gruber, E., *Anesth. Analg.*, 88, 1220-5, 1999.