

MATHEMATISCH PROGRAMMEREN. EEN PROGRAMMA VOOR PC OF MAINFRAME?

Bart van Goudoever¹

Samenvatting

Er zijn Mathematical Programming-programma's beschikbaar voor PC's en voor mainframes. Tussen elk programma bestaan natuurlijk verschillen maar, er bestaat een duidelijk onderscheid tussen de MP-programmatuur voor mainframes en die voor PC's. Dit onderscheid wordt besproken aan de hand van twee programma's. Hierbij komen de mogelijkheden, de gebruikersvriendelijkheid en de snelheid van het programma aan de orde. Conclusie is dat in praktijksituaties PC-programma's bij kleine en middelgrote problemen goed voldoen, ondanks het feit dat een programma voor een mainframe de problemen veel sneller oplost. Voor bijzondere categorieën of grote MP-problemen moet een mainframe MP-pakket worden gebruikt.

¹Drs. B. van Goudoever, Rijksuniversiteit Groningen, Economische Faculteit, Postbus 800, 9700 AV Groningen, tel. 050-633394/637052.

1. Inleiding

Voor bijna elke toepassing op computergebied zijn programma's op PC's en mainframes beschikbaar. Bij het maken van een keuze welk programma voor een bepaald probleem het meest geschikt zou zijn, moet dus eigenlijk ook de selectie van het type hardware betrokken worden. Er zijn belangrijke verschillen tussen programmatuur op PC's en die op mainframes. In dit artikel wordt een vergelijking gemaakt tussen een MP-programma voor mainframes en een MP-programma voor een PC's. Daarmee kan een indruk worden verkregen hoe beide programma's zich tot elkaar verhouden. Beide programma's zijn 'goed' binnen hun soort, alhoewel dat oordeel natuurlijk afhankelijk is van de mate waarin men bepaalde eigenschappen van de verschillende programma's wil meewegen. Belangrijker zijn echter de verschillen tussen de programma's als gevolg van de beperkingen van de hardware waarop ze draaien. De twee besproken programma's geven daarvan een illustratie.

In paragraaf 2 wordt kort ingegaan op enkele technieken van mathematisch programmeren (MP). In paragraaf 3 wordt de gebruikte hard- en software besproken. In de paragrafen 4 en 5 worden de programma's onderling vergeleken.

2. Technieken van MP

Een onderdeel van mathematisch programmeren is lineair programmeren (LP). Hierbij moet een lineaire doelstellingsfunctie worden gemaximaliseerd, onder een aantal (ook lineaire) beperkingen. In vector-notatie ziet een LP-probleem er als volgt uit:

$$\begin{array}{ll} \max & cx \\ \text{zdd} & Ax \leq b \\ & x \geq 0 \end{array}$$

(max staat voor maximaliseer, zdd staat voor zodanig dat).

Hierbij bestaan b en c uit vectoren van n vaste coëfficiënten, is x een vector van n variabelen, en bestaat A uit een matrix van $m \times n$ vaste coëfficiënten. In zo'n geval bestaat het probleem uit n beslissingsvariabelen, terwijl er m restricties zijn. De functie cx heet de doelstellingsfunctie.

Bij de meeste problemen bestaat de matrix A grotendeels uit elementen die gelijk zijn aan nul. Om aan te geven hoe groot een bepaald LP-probleem is wordt, naast

het aantal variabelen en restricties, ook vaak het aantal elementen in de matrix A dat ongelijk is aan 0 aangegeven. Dat is het aantal niet-nul elementen.

Er is een algoritme om een oplossing voor dit probleem te vinden. Dit algoritme heet de simplex-methode (zie bijvoorbeeld [1]). Deze iteratieve methode bestaat uit een aantal stappen, waarvan er een paar net zolang worden herhaald totdat de beste oplossing is gevonden. Hierbij wordt, uitgaande van een bepaalde waarde van x die voldoet aan alle restricties, steeds gekeken of er nog een betere waarde van x is te vinden. Dat wil zeggen een waarde van x waarbij nog steeds aan de restricties wordt voldaan maar waarbij de doelstellingsfunctie cx een hogere waarde heeft.

De simplex-methode kan worden versneld door van een geschikte startwaarde van x uit te gaan. Er zijn technieken om zo'n startwaarde te vinden. Een dergelijke techniek staat soms bekend als het Crash-algoritme.

Onder Mathematical Programming worden, naast Lineair Programming (LP), onder andere ook Mixed Integer Programming (MIP), Integer Programming (IP) en Non-Linear Programming (NLP) onderscheiden. In MIP-problemen komen naast reële variabelen ook variabelen voor die alleen een geheeltallige waarde mogen aannemen. In IP-problemen moeten alle variabelen geheeltallig zijn, terwijl in NLP-problemen de doelstellingsfunctie en/of een aantal restricties niet lineair hoeven te zijn.

Een bijzondere categorie van MIP problemen zijn de zogenaamde Mixed Binary Programming (MBP)-problemen. Hierbij mogen de geheeltallige variabelen alleen de waarde 0 of 1 aannemen. Alhoewel dit een aanzienlijke beperking lijkt valt dat in de praktijk wel mee, veel MIP-problemen zijn MBP-problemen. Afgezien daarvan kunnen ook geheeltallige variabelen met een MBP-programma worden gemodelleerd door een aantal binaire variabelen met steeds oplopende machten van 2 te vermenigvuldigen en die produkten vervolgens te sommeren. Ook omdat programma's om MBP-problemen op te lossen makkelijker te maken zijn dan programma's voor MIP-problemen staan veel PC-programma's alleen MBP toe.

MIP-problemen worden opgelost met behulp van een branch-and-bound techniek. Het simplex-algoritme maakt hier deel van uit. De techniek komt erop neer dat eerst het MIP-probleem wordt opgelost als ware het een LP-probleem. Vervolgens wordt, via een of ander algoritme, een variabele gekozen die geheeltallig moet zijn

maar die nog geen geheeltallige waarde heeft. Laat dit variabele x_i zijn. Het integer gedeelte van x_i wordt $\text{int}[x_i]$ genoemd. Van het bestaande probleem worden nu twee nieuwe problemen gemaakt door er (twee keer) één nieuwe vergelijking aan toe te voegen. Dit zijn respectievelijk de vergelijkingen:

$$x_i \leq \text{int}[x_i] \quad \text{en} \quad x_i \geq \text{int}[x_i] + 1$$

Daarna wordt, via een of ander algoritme, één van beide problemen gekozen en opnieuw als LP-probleem opgelost. Vervolgens wordt de hierboven beschreven procedure een aantal malen herhaald. Normaal gesproken zullen er op een gegeven moment een aantal oplossingen zijn gevonden die voldoen aan de eisen van geheeltalligheid. Aan de hand van een bepaald stopcriterium wordt de beschreven procedure vervolgens beëindigd en wordt de beste tot dan toe gevonden oplossing gepresenteerd.

Enkele gangbare stopcriteria zijn:

- Stop als er een vast aantal geheeltallige oplossingen zijn gevonden,
- stop na een bepaalde periode rekenen,
- stop als de beste tot dan toe gevonden oplossing niet meer dan een bepaald percentage of een bepaalde absolute hoeveelheid van de continue oplossing afwijkt,
- stop als de beste tot dan toe gevonden geheeltallige oplossing niet meer kan worden verbeterd.

De kwaliteit van het branch-and-bound algoritme, dat wil zeggen de snelheid waarmee het algoritme tot een goede of zelfs de beste geheeltallige oplossing komt, is met name afhankelijk van de keuze van x_i en van het model dat vervolgens wordt opgelost. IP- en NLP-problemen worden met andere technieken opgelost. Omdat ze in de praktijk veel minder worden gebruikt zullen deze hier niet worden behandeld.

3. De selectie van programma's en apparatuur

Het oplossen van MP-problemen is erg rekenintensief, vergelijkbaar met toepassingen op het terrein van animatie of Computer Aided Design. Zowel voor mainframes als voor PC's zijn er diverse programma's beschikbaar die MP-problemen kunnen oplossen. Vooral op PC's verschillen deze programma's nogal in kwaliteit.

Een onderlinge vergelijking tussen een aantal LP-programma's voor PC's is te vinden in [2], [3] en [4].

De PC-programmatuur is onder te verdelen in twee groepen. Sommige programma's werken alleen samen met een spreadsheet (zoals bijvoorbeeld What's Best en Optimal Solutions), andere programma's hebben geen spreadsheet nodig (zoals bijvoorbeeld PcProg en XA). Veel van de programma's in de laatste groep kunnen overigens wel een spreadsheet-file waarin het model is gespecificeerd lezen. Beperkingen of bezwaren bij de programma's voor PC's zijn dat ze in sommige gevallen erg traag zijn en in sommige gevallen slechts kleine modellen kunnen oplossen. Hiervan afgezien is ook de functionaliteit van PC-programma's niet erg groot, er zijn er maar weinig die zowel LP-, MIP-, IP- en NLP-problemen kunnen oplossen. PcProg [3] is hierop een uitzondering, maar dit programma kent duidelijk zijn beperkingen in de maximale omvang van het probleem dat kan worden opgelost.

Omdat we met name geïnteresseerd waren in de vraag hoe de programma's functioneerden bij middelgrote problemen (tenminste honderd variabelen en restricties en niet meer dan 5000 niet-nul elementen) is uiteindelijk XA geselecteerd. In vergelijking tot andere MP-programma's zoals PcProg, Lindo, What's Best en Optimal Solutions kan XA redelijk grote problemen in een redelijk tempo oplossen. XA kan problemen aan waarbij de matrix A maximaal uit 1500 rijen en 8170 kolommen bestaat, waarbij 32000 elementen in de matrix ongelijk aan 0 mogen zijn. Bezwaar van XA is dat het alleen LP- en MBP-problemen kan oplossen. Het programma kost ongeveer f1500,-.

Programma's op het mainframe ontlopen elkaar niet veel in de geboden mogelijkheden. In de meeste gevallen kunnen vele soorten MP-problemen worden opgelost. Omdat de beschikbare hoeveelheid werkgeheugen bij een mainframe veel groter is dan een PC kunnen grote problemen worden opgelost. Daarnaast bieden MP-programma's op het mainframe veel extra numerieke technieken (zoals bijvoorbeeld Crash) om snel een oplossing te vinden. Als de vereiste MP-mogelijkheden aanwezig zijn zal de koper eerder geïnteresseerd zijn in de prijs dan in welke technieken verder precies aanwezig zijn. De prijs van zo'n programma ligt al gauw tussen de f10.000,- en f100.000,-. Dit artikel beschrijft het programma Sciconic, dat beschikbaar is op tal van machines zoals die van IBM, Unisys, CDC en Digital. Sciconic kost ongeveer f50.000,-.

De werking van XA is getest op een PC met een 80386 processor, voorzien van een 80387 coprocessor, met een kloksnelheid van 25 MHz. Het gebruik van een coprocessor scheelt ongeveer een factor 5 in de tijd die nodig is om het probleem op te lossen. Sciconic is geïmplementeerd op de Cyber 962 van CDC. De Cyber 962 is een machine met 64 Mb intern geheugen en voert 15 MIPS uit. De toegangssnelheid van de processor is 300 nanoseconden. Van beide computers kan worden gezegd dat het misschien niet de allersnelste, maar toch wel zeer snelle computers in hun soort zijn.

4. De vergelijking

MP-programma's zijn aan de hand van drie criteria vergeleken. Dat zijn de functionaliteit die ze bieden in combinatie met de technieken die ze ondersteunen, de gebruikersvriendelijkheid en de tijd die het kost om een oplossing te vinden. Het laatste is natuurlijk mede afhankelijk van de computer waarop het programma is geïmplementeerd en de technieken die in het programma zijn opgenomen.

4.1. Funktionaliteit en technieken

Eén van de grootste problemen bij PC-programmatuur is dat er moet worden gewoekerd met de geheugenruimte. Daarom maken de ontwikkelaars veelal programma's die alleen LP en MIP of MBP problemen kunnen oplossen, waarin bovendien slechts een beperkt aantal technieken zijn opgenomen. Dan blijft er nog genoeg geheugenruimte over om middelgrote problemen op te lossen. Er is één uitzondering, het programma PC-Prog (zie [3]). Dit programma, dat vooral bedoeld is voor gebruik binnen het onderwijs, kan wel IP- en NLP-problemen oplossen. Het aantal extra technieken is ook bij dit programma klein, terwijl duidelijk een concessie is gedaan aan de maximaal toegestane omvang van het probleem.

Ook XA kent maar een beperkt aantal extra technieken. Naast LP-problemen kunnen alleen MBP-problemen worden opgelost. Er zijn 4 mogelijke branch-and-bound strategieën, verder kan de gebruiker niet veel meer van het oplossingsproces beïnvloeden.

Bij het mainframe-programma ligt dit duidelijk anders. Sciconic kan LP-, MIP, IP en sommige NLP-problemen oplossen. Bij MIP-problemen kent Sciconic de mogelijk-

heid om groepen van variabelen te specificeren, de zogenaamde S1 en S2 sets. Een belangrijk kenmerk van zo'n groep is dat er maar één of twee variabelen ongelijk aan hun laagst toegestane waarde mogen zijn. Het oplossingsproces van Sciconic kan op vele manieren worden beïnvloed en gestuurd, waaronder de Crash-techniek. Er zijn vele branch-and-bound technieken beschikbaar, die tussentijds kunnen worden bijgestuurd.

4.2. Gebruikersvriendelijkheid

MP-programma's blinken meestal niet uit in gebruikersvriendelijkheid. De moeilijkheid zit vooral in de manier waarop het model moet worden gespecificeerd. Een matrix van $100 * 100$ is nu eenmaal niet overzichtelijk. De meeste programma's, zowel voor PC als mainframe, accepteren invoer in MPS-formaat. Dit is een standaard formaat (alhoewel er vele dialecten bestaan) uit het ponskaart-tijdperk die erop neer komt dat de elementen ongelijk aan 0 in de matrix A stuk voor stuk worden weergegeven. Grote modellen worden zelden opgesteld in MPS-formaat. Meestal ontwikkelt of koopt men dan een MPS-generator, een programma dat van één of andere beter leesbare modeltaal een MPS-model maakt.

Is bij mainframe programmatuur (ook bij Sciconic) het MPS formaat meestal het enige toegestane formaat, PC-programma's staan vaak ook nog andere formaten toe. In XA bijvoorbeeld mogen modellen naast het MPS-formaat worden gespecificeerd in een spreadsheet of in een modeltaal. PC-programma's zijn daarmee wat vriendelijker in gebruik dan mainframe programma's. De gebruiker heeft meer mogelijkheden om het model te specificeren. Daarnaast kan men meestal beter overweg met een editor of tekstverwerker op een PC dan zo'n programma op een mainframe. Een klein bijkomend voordeel van een PC is dat het operating system van een PC makkelijker is dan het operating system van een mainframe.

De gebruikersvriendelijkheid van de MP-programma's zelf zal elkaar niet veel ontlopen. Voor het oplossen van een probleem hoeven slechts een beperkt aantal opdrachten te worden gegeven. Een mainframe-programma heeft meer mogelijkheden, zodat in de meeste gevallen iets meer opdrachten moeten worden gegeven in vergelijking tot een PC-programma.

Voor de uitvoer geldt eigenlijk weer hetzelfde als bij de invoer. De uitvoer bestaat uit een bestand waarin de oplossing wordt gegeven. Deze moet in de meeste gevallen nog verder worden bewerkt, waarbij PC's wat handiger in het

gebruik zijn. Bovendien kunnen diverse andere PC-programma's makkelijk de uitvoerbestanden lezen, zodat de oplossing zonder al te veel conversieslagen kan worden opgenomen in een database, een tekstbestand of een spreadsheet.

Een aantal van de bezwaren tegen een mainframe wordt weggenomen als gebruik wordt gemaakt van een hardware-configuratie waarbij het model met een PC wordt gespecificeerd, en vervolgens verzonden wordt naar het mainframe om daar met het MP-programma te worden opgelost. De resultaten kunnen weer worden verzonden naar de PC en daar verder worden verwerkt.

4.3. De snelheid van oplossen

Er zijn twee redenen om aan te nemen dat mainframe-programma's sneller een MP-probleem oplossen dan een PC-programma. Ten eerste is een mainframe een snellere computer dan een PC. Ten tweede beschikt een mainframe MP-programma over meer technieken op een MP-probleem op te lossen.

Om eens een indicatie te krijgen van de snelheid van Sciconic op de Cyber in vergelijking tot XA op een 80386 machine zijn een aantal MBP-problemen op beide machines opgelost. De omvang van de problemen is te vinden in tabel 1.

tabel 1: De omvang van de problemen.

	aantal variabelen	aantal restricties	aantal niet-nul	aantal binair
Frisjes	109	157	427	30
Wolf	136	177	554	136
Kaars	186	222	597	18
Karren	200	191	637	13
Bitbulk	169	235	654	12
Finall	1094	946	3097	73
Final2	1454	1256	4121	95

Alle problemen op finall en final2 na zijn verschillend in structuur. Het probleem wolf betreft een theoretisch probleem, alle andere problemen beschrijven planningproblemen op het terrein van produktie, marketing en financiering in organisaties.

De tijd die het kostte om de problemen op te lossen is gegeven in tabel 2. Om te illustreren dat extra technieken de rekentijd kunnen verlagen zijn bij Sciconic de problemen zowel met als zonder het Crash-algoritme opgelost. Op voorhand is vaak niet te zeggen welk branch-and-bound algoritme het snelst zal zijn. Vandaar dat dikwijls wordt gekozen voor het algoritme dat in de meeste gevallen het

snelst zal zijn. In deze test is dat ook gedaan, het algoritme dat door de programmaontwikkelaar wordt aanbevolen is gebruikt.

tabel 2: De benodigde rekentijd in seconden.

	rekentijd in sec.			Verhouding	
	PC	Mainframe		PC / Mainframe	
		met crash	zonder crash	met crash	zonder crash
frisjes	1125	2.4	15.6	469	72
wolf	6060	2.2	1.8	2705	3443
kaars	356	4.4	6.2	81	57
karren	38	2.0	1.9	19	20
bitbulk	342	34.1	33.1	10	10
final1	4440	35.9	85.2	124	52
final2	48720	36.9	599.4	1320	81

Voor het mainframe is de rekentijd in CPU-seconden gemeten. De doorlooptijd (het tijdsverschil tussen het wegzenden van het probleem en het terugkrijgen van het antwoord) is vele malen hoger. De doorlooptijd is in het algemeen afhankelijk van de loading van het systeem, de prioriteit die de gebruiker en het programma hebben en de hoeveelheid CPU-seconden dat nodig is om het probleem op te lossen. De verhouding tussen het aantal CPU-seconden en doorlooptijd ligt voor elke programma/computer/gebruiker anders. Het meten van de doorlooptijd, voor de gebruiker een belangrijk aspect, is daarom voor deze test geen zinnige maat. Elke gebruiker kan voor zichzelf een schatting maken van de verhouding tussen rekentijd en doorlooptijd, en daar eventueel de testresultaten mee corrigeren.

Uit tabel 2 blijkt dat Sciconic op de Cyber de oplossing aanzienlijk sneller vindt dan XA op de PC. Het crash-algoritme levert, zeker bij de grote problemen, een aanmerkelijke tijdswinst op.

Het probleem Wolf springt er duidelijk tussen uit. De verklaring voor het grote verschil in rekentijd dat nodig is bij het vinden van een oplossing ligt grotendeels in het efficiëntere branch-and-bound algoritme van Sciconic. Als bij XA een ander algoritme wordt gebruikt kan de benodigde rekentijd met 75% worden teruggebracht.

5. Vergelijking en conclusies

Bij vergelijking van beide programma-categorieën ontstaat het volgende beeld. MP-programma's voor mainframes ondersteunen vele soorten van MP, en kennen binnen elke groep veel meer extra technieken. Een derde belangrijk onderscheid is dat een mainframe-programma veel grotere problemen kan oplossen.

Door de extra mogelijkheden in het programma en omdat mainframes snellere computers zijn, vindt een mainframe-programma de oplossing aanzienlijk sneller. Gezien het grote relatieve verschil in benodigde rekentijd tussen de testproblemen is het niet goed mogelijk een uitspraak te doen hoeveel malen sneller het mainframe in vergelijking tot een PC bij deze twee programma's is. Wat wel gezegd kan worden is dat bij deze problemen het mainframe in totaal 118 seconden nodig heeft, terwijl de PC er ruim 60.000 seconden over doet. Het mainframe met Sciconic was voor deze problemen dus ruim 500 keer zo snel. Het verschil zal voor een groot deel toe te schrijven zijn aan de hardware, maar ook verschillen in de software zullen bijdragen aan het snelheidsverschil.

Een PC-programma is gebruikersvriendelijker. Het model is makkelijker op een PC te specificeren dan op een mainframe, en de uitvoer kan makkelijker worden verwerkt op een PC. De tijd die de gebruiker kwijt is aan het ontwikkelen van het model en het analyseren van de uitkomsten is op een PC korter dan op een mainframe. Hoeveel het precies zal schelen is afhankelijk van het aantal malen dat het model moet worden gewijzigd, de ervaring die de gebruiker heeft met de apparatuur en programmatuur en dergelijke.

Voor de kosten voor het uiteindelijke oplossen van het model moet onderscheid worden gemaakt tussen vaste en variabele kosten. De vaste kosten worden hier gedefinieerd als de kosten voor de software. XA kost ongeveer f1500, Sciconic ongeveer f50.000. De variabele kosten zijn gedefinieerd als de kosten per rekenseconde. De PC die is gebruikt kost ongeveer f10.000,-, het mainframe f5 miljoen. Een seconde rekenen is op de Cyber dus ongeveer 500 keer zo duur als een seconde rekenen op de PC, (onder een aantal aanvullende aannames). Deze factor 500 komt overeen met het relatieve snelheidsverschil bij de testproblemen, zodat de variabele kosten bij deze problemen ongeveer even hoog zijn.

Samenvattend kan worden gesteld dat mainframe MP-programma's veel meer mogelijkheden hebben en grotere problemen aankunnen. PC MP-programma's zijn gebruikers-

vriendelijker. Met PC-programma's kunnen modellen sneller worden gespecificeerd, terwijl mainframe programma's de modellen sneller oplossen. Als men alleen kleine of middelgrote LP- of MBP-problemen wil oplossen dan zal een PC-programma in het algemeen uitstekend voldoen.

Literatuur:

[1] Hillier and Lieberman, Introduction to Operations Research, Holden-Day Inc. New York, 1969.

[2] De Personal Computer-gids, Samsom, Alphen a/d Rijn, 1988 (losbladige uitgave).

[3] Rutten, W.G.M.M., Enige in Nederland verkrijgbare PC-software pakketten voor mathematische programmering, in: Kwantitatieve Methoden, no 31, jrg 10, pag. 171-184, juni, 1989.

[4] Sharda, R., The state of the Art of Lineair Programming on Personal Computers, in: Interfaces, no 18, pag 49-58, jul-aug, 1988.