'n Eksperimentele ondersoek na 'n nuwe multipleksmetode vir lineêre programmering

J.A. Snyman en M. van Rooyen

Departement Wiskunde en Toegepaste Wiskunde, Universiteit van Pretoria, Pretoria 0002

UITTREKSEL

Die onlangse interne en iteratiewe metode van Karmarkar het 'n hernude belangstelling in ouer alternatiewe metodes, verskillend van die simpleksmetode, gewek. Hier word 'n nuwe multipleks- en meetkundige metode, wat sekere ooreenstemmings met die ouer metodes toon, voorgestel en geïmplementeer. In dié metode word die oplossing verkry deur 'n gradiëntbaan deur die binnekant van die moontlike gebied, en deur subruimtes van kleiner dimensie wat ooreenstem met die begrensende hipervlakke van die moontlike gebied, te volg. Die baan beweeg vanaf 'n aanvanklike moontlike punt deur 'n reeks lineêre stappe na 'n verteks van die politoop wat deur die begrensings gedefinieer is. Alhoewel dit soortgelyk skyn te wees, verskil die huidige metode tog fundamenteel van Rosen se gradiëntprojeksiemethode in dié opsig dat die agtereenvolgende soekrigtings verkry word vanaf die gradiënte van gereduseerde probleme van kleiner dimensie. Wanneer hierdie rigtings na die oorspronklike ruimte getransformeer word, stem dit nie noodwendig met die gradiëntprojeksierigtings ooreen nie. As 'n verteks bereik word, bereken die nuwe algoritme of dit optimaal is al dan nie. Dit word gedoen deur 'n eenvoudige perturbasieprosedure toe te pas waardeur geperturbeerde punte, as 'n neweproduk van die berekende baan na die verteks, gegenereer word. Indien nie optimaal nie, gaan die algoritme voort deur oor te begin vanaf 'n geperturbeerde punt (op 'n geskikte rand) met groter funksiewaarde, en die baan word voortgesit totdat 'n nuwe verteks bereik word. Dit sal blyk dat die resultate van hierdie eksperimentele ondersoek na dié betrokke metode belowend is, aangesien die metode suksesvol op 'n verskeidenheid toetsprobleme toegepas is.

ABSTRACT

An experimental investigation of a new multiplex method for linear programming

Karmarkar's recent internal and iterative method for linear programming problems has resulted in a renewed interest in some older alternatives, other than the simplex method. Here a new multiplex and geometric method, which has some features in common with the older methods, is proposed and implemented. In this method the solution is found by following a gradient path through the interior of the feasible region and through subspaces of reduced dimension corresponding to the bounding hyper-surfaces of the feasible region. The path moves from an initial feasible point through a sequence of linear steps to a vertex of the polytope defined by the constraints. Although similar, the current method differs fundamentally from Rosen's gradient projection method in that the successive search directions are obtained from the gradients of reduced problems of lesser dimension. These directions, when translated to the original space, do not necessarily correspond to the gradient projection directions. Once a vertex has been reached the new algorithm determines whether or not it is optimal by applying a simple perturbation procedure for which the perturbed points are generated as a by-product of the computed path to the vertex. If not optimal the algorithm proceeds by restarting from a perturbed point (on a suitable edge) with increased function value and the path is continued until the next vertex is reached. It is shown that the results of this experimental investigation of this method is promising since it has successfully been applied to a variety of test problems.

1. INLEIDING

'n Groot groep praktiese probleme, byvoorbeeld dié wat voorkom by die bepaling van militêre strategieë, interindustriële ekonomie en by sosiale- en energiebehoefte, kan in terme van Lineêre Programmering (LP) geformuleer word. Onlangs is groot belangstelling gewek deur die publikasie van Karmarkar¹ se nuwe polinoomtydalgoritme vir lineêre programmeringsprobleme. Die aanspraak is gemaak dat dié metode, anders as die eerste gepubliseerde polinoomtydalgoritme van Khachian,² vir praktiese probleme baie vinniger is as die langgevestigde en bekende simpleksmetode wat deur Dantzig³ in die laat 1940's ontwikkel is. Alhoewel dit van die begin af reeds bekend was dat laasgenoemde metode oor die ergste graad van kombinatoriese ingewikkeldheid beskik, is dit in die praktyk verbasend suksesvol. Nie

alleen neem die aantal iterasies wat vir die oplos van 'n probleem vereis word, ongeveer lineêr met die dimensie van die probleem toe nie, maar word ook relatief min werk deur 'n tipiese iterasie vereis. Die skynbaar onbevredigende aspek van die simpleksmetode is dat dit langs die een-dimensionele rande van die toelaatbare oplossingsgebied na die optimale oplossingspunt beweeg. In teenstelling hiermee blyk dit uit 'n uiteensetting⁴ van Karmarkar se metode dat hy, deur die toepassing van slim transformasies, deurgaans daarin slaag om die optimale oplossing vanuit 'n sentrale interne punt te benader. Hierdie waarneming het as motivering gedien om meer konvensionele en meetkundige interne metodes te ondersoek. Dit blyk dat reeds vanaf die vroeë ontwikkelingsjare van die simpleksmetode pogings aangewend is om praktiese lineêre programmeringsmetodes te

ontwikkel wat die binnekant van die toelaatbare gebied oorkruis.^{5,6} Hierdie metodes was egter nie baie suksesvol nie en dit blyk dat daar gevolglik, gedurende die afgelope 25 jaar, min ernstige aandag aan sodanige beskouings gegee is.

In hierdie studie word 'n nuwe interne metode voorgestel en beproef. Die oplossing word verkry deur 'n gradiëntbaan deur die binnekant en langs die begrensende hipervlakke van die toelaatbare gebied te volg. Die baan beweeg vanaf 'n aanvanklike toelaatbare beginpunt deur 'n reeks lineêre stappe na 'n verteks van die toelaatbare politoep wat deur die begrensings gedefinieer word. So 'n baan, meer kompleks in vergelyking met die randbane van die simpleksmetode, noem ons multipleksbane en die metode 'n multipleksmetode. Dié metode stem in sekere opsigte ooreen met Rosen se gradiëntprojeksiemethode,⁷ maar is tog fundamenteel verskillend. Beide metodes gebruik die gradiënt van die doelfunksie. Rosen se metode werk deurentyd in die oorspronklike ruimte met beweging in die rigting van 'n gradiëntprojeksievektor wat van die aktiewe begrensings afhang. In teenstelling hiermee transformeer die huidige metode, weer eens afhanglik van die aktiewe begrensings, die oorspronklike LP-probleem na verwante gereduseerde LP-probleme in ruimtes van kleiner dimensie. Dit blyk ook dat waar die gradiëntprojeksiebaan uniek is, daar meer as een moontlike oplossingsbaan by die toepassing van die nuwe voorgestelde metode mag bestaan. Die huidige metode toon ook sekere ooreenkomste met die gereduseerde gradiëntmetode van Wolfe⁸ en met die multipleksmetode van Frisch,⁹ maar die formulering en implementering van laasgenoemde metodes skyn tog geheel en al verskillend te wees.

Alhoewel teoretiese beskouings in die ontwikkeling van die nuwe metode gebruik word en die konstruksie van die algoritme op konvergensie dui, is daar geen formele poging om 'n streng teoretiese bewys vir konvergensie te gee nie. Die belangrikste teoretiese eienskappe van die algoritme en 'n aanduiding van 'n konvergensiebewys word egter in bylae A gegee. Die belangrikste klem van hierdie studie is eksperimenteel en die geldigheid van die voorgestelde metode moet dus, in 'n groot mate, aan die hand van eksperimentele toetsresultate gemeet word. Dit sal aangetoon word dat die resultate van hierdie voorlopige eksperimentele ondersoek belowend is, aangesien die metode suksesvol op 'n verskeidenheid toetsprobleme toegepas is.

2. DIE MULTIPLEKSMETODE

2.1 Die lineêre programmeringsprobleem

Beskou die probleem:

$$\text{maksimeer } f(\underline{x}) = \underline{c}'\underline{x} \quad , \quad \underline{c}, \underline{x} \in \mathbb{R}^n ;$$

só dat (1)

$$\underline{g}(\underline{x}) = \underline{A}\underline{x} - \underline{b} \leq \underline{0} \quad , \quad \underline{A} \in \mathbb{R}^{m \times n} \quad , \quad \underline{g} \in \mathbb{R}^m .$$

$\underline{g}(\underline{x}) = [g_1(\underline{x}), g_2(\underline{x}), \dots, g_m(\underline{x})]'$ word die begrensings van die probleem genoem en $f(\underline{x})$ die doelfunksie. Die

konvensionele nie-negatiwiteitsbegrensings $\underline{x} \geq \underline{0}$, ($-\underline{x} \leq \underline{0}$) word in hierdie formulering ingesluit as deel van die begrensings $\underline{g}(\underline{x}) \leq \underline{0}$. Die gebied waarin die begrensings bevredig word, word die toelaatbare gebied genoem en sal deur Ω aangedui word. Die i -de ry van die matriks $\underline{A} = (a_{ij})$ word deur die vektor $\underline{a}_i$ aangedui.

Om die nuwe multipleksalgoritme (MP-algoritme) te beskryf, word daar aanvaar dat 'n optimale oplossing tot (1) bestaan. Verder word aanvaar dat 'n toelaatbare beginpunt, d.w.s. 'n punt in Ω , sê $\underline{x}^1$, bekend is.

2.2 Die optimale stapgrootte

Uit die differensiaalrekening volg dit dat die rigting waarin 'n skalaarfunksie maksimaal toeneem, in die rigting van sy gradiënt is. Die MP-algoritme konstrueer 'n nuwe en beter toelaatbare oplossing, $\underline{x}^2$, deur 'n stap in die rigting van die gradiënt vanaf $\underline{x}^1$ te neem. Aangesien $\nabla f = \underline{c}$ in dié geval 'n konstante vektor is, word $\underline{x}^2$ gegee deur

$$\underline{x}^2 = \underline{x}^1 + h\underline{c} \quad (2)$$

waar die staplengteparameter h so groot moontlik gekies word, maar so dat $\underline{x}^2$ steeds in Ω lê. Die optimale stapgrootte is 'n stap in die rigting van $\underline{c}$ tot op die rand van Ω , wat ons deur $\partial\Omega$ aandui. $\partial\Omega$ word deur 'n deelversameling van aktiewe begrensing beskryf. Hiermee word die volgende bedoel: daar bestaan indeksversamelings Λ_k elk van k elemente sodat $\partial\Omega$ die vereniging is van

$$\partial\Omega_k = \{ \underline{x} / g_i(\underline{x}) = 0, i \in \Lambda_k \} \quad (3)$$

waar $\Lambda_k \in \{1, 2, \dots, m\}$ en $\underline{x} \in \Omega$, en wat ooreenstem met die verskillende begrensende hipervlakke (van dimensie $n-1$ tot 0).

Die probleem is egter dat die verskillende Λ_k onbekend is en dus het ons nie 'n karakterisering van $\partial\Omega$ nie. Hierdie probleem word omseil deur daarop te let dat $V_i = \{ \underline{x} / g_i(\underline{x}) = 0 \}$ bloot 'n $(n-1)$ -dimensionele hipervlak in $\mathbb{R}^n$ is. Daar hoef dus net gestap te word totdat die eerste vlak getref word. Die afstand na V_i , in eenhede van $\|\underline{c}\|$, is dié h_i só dat $g_i(\underline{x}^1 + h_i\underline{c}) = 0$. h_i kan eksplisiet opgelos word, indien $\underline{c}'\underline{a}_i$ ongelyk aan nul is. Dit lewer:

$$h_i = -(\underline{c}'\underline{a}_i)^{-1} g_i(\underline{x}^1) \quad (4)$$

Deur dus 'n h vir elke begrensing te bereken en die kleinste daaruit te kies, sal die verlangde stapgrootte tot by $\partial\Omega$ verkry kan word. Let egter daarop dat, om te verseker dat f toeneem, slegs nie-negatiewe h 's in ag geneem mag word. Gevolglik, omdat $g(\underline{x}^1) \leq \underline{0}$, word alle gevalle waar $\underline{c}'\underline{a}_i < 0$, geïgnoreer. Dit sluit ook die geval $g_i(\underline{x}^1) = 0$ met $h_i = 0$ in, aangesien $\underline{c}'\underline{a}_i < 0$ impliseer dat toename weg van die begrensing in die rigting van ∇f plaasvind en die begrensing dus nie aktief bly nie. As $\underline{c}'\underline{a}_i = 0$, is (4) ongedefinieerd en beskou ons die twee moontlike situasies, nl. $g_i(\underline{x}^1) < 0$ of $g_i(\underline{x}^1) = 0$ (omdat $\underline{x}^1 \in \Omega$). Indien die eerste situasie hom voordoën, beteken dit dat daar slegs parallel aan, maar weg van die vlak, beweeg kan word. Ons het hier effektiewelik dat $h = \infty$, sodat hierdie begren-

sing geïgnoreer kan word. Indien die tweede situasie voorkom, lê $\underline{x}^1$ in V_i en het ons effektiewelik dat die begrensing aktief is, aangesien enige moontlike verdere stap in die vlak V_i sal plaasvind. Die begrensing word egter nie aktief afgedwing nie en dus word dit ook geïgnoreer. Die volgende konvensie word aanvaar:

$$\begin{aligned} h_i &= -(\underline{c}'\underline{a}_i)^{-1} g_i(\underline{x}^1) \quad \text{as } \underline{c}'\underline{a}_i > 0 \\ h_i &= -1 \quad \text{andersins (d.i., ignoreer as } \underline{c}'\underline{a}_i \leq 0) \end{aligned} \quad (5)$$

Die verlangde stapgrootte tot by $\partial\Omega$ word vervolgens gegee deur

$$h = \text{minimum } \{h_i/h_i \geq 0\} \quad (6)$$

Bostaande prosedure lewer ten minste een begrensing, sê g_k , waarvoor geld dat $g_k(\underline{x}^2) = 0$ en dié begrensing dus aktief is.

2.3 Die gereduseerde LP-probleem

Die MP-algoritme vereis nou dat die volgende stap in die vlak V_k moet plaasvind. Dit word afgedwing deur die vergelyking $g_k = 0$ te gebruik om een veranderlike uit (1), deur middel van substitusie, op die volgende wyse te elimineer: Beskou die aktiewe begrensing

$$g_k(\underline{x}) = a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n - b_k = 0 \quad (7)$$

Een van die veranderlikes, sê x_a , kan dus in terme van die oorblywende $n-1$ onafhanklike veranderlikes

$$\underline{x}_0 = [x_1, x_2, \dots, x_{a-1}, x_{a+1}, \dots, x_n]$$

opgelos word: (8)

$$x_a = -[a_{k1}x_1 + \dots + a_{kn}x_n - b_k]/a_{ka}$$

Deur nou x_a te vervang in $f = \underline{c}'\underline{x}$ en $g(\underline{x})$ van die oorspronklike probleem (1), verkry ons die gereduseerde LP-probleem van dimensie $n-1$ met een minder begrensing:

$$\text{maksimeer } f^*(\underline{x}_0) = \underline{c}^*\underline{x}_0 + K$$

só dat (9)

$$g_j(\underline{x}_0) = \underline{a}_j^*\underline{x}_0 - b_j \leq 0, \quad j \neq k$$

waar

$$\begin{aligned} c_i^* &= c_i - c_a a_{ki}/a_{ka}, \quad i \neq a \\ K &= -c_a b_k/a_{ka} \\ a_{ji}^* &= a_{ji} + a_{jk} a_{ki}/a_{ka} \quad \text{en} \\ b_j^* &= b_j + a_{jk} b_k/a_{ka} \end{aligned}$$

Nou word (9) as 'n LP-program in eie reg beskou en daar word weer soos vantevore 'n stap, die keer in die rigting $\nabla f^* = \underline{c}^*$ in R^{n-1} , geneem. Dit lewer weer eens nog 'n aktiewe begrensing, sê g_m . Die vraag is nou hoe hierdie nuwe inligting benut moet word.

Die volgende strategie word nou voorgestel. As die laaste stapgrootte nul was, word nog 'n veranderlike uit (9) geëlimineer, aangesien ons aanvaar dat twee begrensings, nl. g_k en g_m , nou gelyktydig aktief is. Dit lewer nou op sy beurt 'n nuwe gereduseerde LP-probleem van dimensie $n-2$ met $m-2$ begrensings. Indien die laaste stapgrootte nie nul was nie en 'n stap geneem is, word nou weer slegs die nuwe aktiewe

begrensing gebruik om slegs een veranderlike uit (1) te elimineer om weer 'n nuwe gereduseerde LP-probleem, weer van dimensie $n-1$ en $m-1$ begrensings, te gee. Dit word gedoen omdat g_k nou moontlik nie langer aktief mag wees nie. Vervolgens, in beide gevalle, word daar nou weer stapgroottes bereken wat, afhangende daarvan of die minimum toelaatbare stapgrootte gelyk aan nul of groter as nul is, onderskeidelik sal lei tot die verdere eliminasië van nog 'n veranderlike of net die eliminasië van 'n enkele veranderlike deur die instelling van die nuutste begrensing wat aktief geword het. Hierdie prosedure word nou herhaaldelik deurgevoer met die hoop dat as daar aktiewe lineêre onafhanklike begrensings op die manier gevind is, die oplossing van die lineêre stelsel van aktiewe begrensings die gevraagde oplossing is tot probleem (1) en die algoritme termineer. Uit voorlopige eksperimente het dit egter geblyk dat hierdie veronderstelling foutief is. Alhoewel die algoritme dikwels met die eerste terminasië die optimale punt mag bereik, was daar gevalle waar vroegtydige terminasië by 'n nie-optimale verteks plaasvind. Om dit te verhoed, moet die algoritme voorsiening maak vir 'n toets vir optimaliteit. Indien die verteks dan nie-optimaal is nie, moet 'n perturbasie-prosedure uitgevoer word na 'n nie-verteksposisie met groter funksiewaarde, vanwaar die algoritme verder kan vorder. Sodoende word 'n reeks multi-pleksbane, waarvan elk by 'n verteks eindig, voortgebring.

2.4 Die MP-algoritme

Vervolgens word die volle besonderhede van die verskillende stappe wat deur die MP-algoritme uitgevoer word, hieronder aangegee.

Stap 1. Aanvaar dat 'n toelaatbare beginpunt, sê $\underline{x}^1$, bekend is. Stel nou $\underline{x}^2 \leftarrow \underline{x}^1 + h\underline{c}$ met h bereken volgens (5). Dit lewer een aktiewe begrensing, sê g_k . Stel $w_1 \leftarrow k$, $p \leftarrow 1$, $i \leftarrow 1$ en stel vlaggie $v \leftarrow 0$.

Stap 2. Stel $i \leftarrow i + 1$.

Stap 3. Aanvaar nou 'n stel aktiewe begrensings $W = \{w_s\}$, $s = 1, 2, \dots, p$; by toelaatbare punt $\underline{x}^1$.

Stap 4. Elimineer p veranderlikes uit (1) d.m.v. die p vergelykings wat die p aktiewe begrensings beskryf, deur die volgende te doen.

Skryf die stel p aktiewe begrensings, deur geskikte herrangskikking van veranderlikes, in die volgende vorm:

$$P\underline{x}_a + Q\underline{x}_0 = \underline{d} \quad (10)$$

waar P 'n nie-singuliere $p \times p$ matriks is (dit kan altyd gevind word indien die aktiewe begrensings lineêre onafhanklik is) en Q 'n $p \times (n-p)$ matriks, met $\underline{d}$ 'n p -vektor. Net soos in die eenvoudiger geval van slegs een aktiewe begrensing, (7) en (8), los ons nou ook op vir p afhanklike veranderlikes i.t.v. die $n-p$ onafhanklike veranderlikes:

$$\underline{x}_a = P^{-1}\underline{d} - P^{-1}Q\underline{x}_0 \quad (11)$$

en vervang dit in $f = \underline{c}'\underline{x}$ en begrensings $g(\underline{x})$ van die oorspronklike probleem (1). Dit lei tot die meer algemene gereduseerde probleem:

$$\text{maksimeer } f^*(\underline{x}_0) = \underline{c}^* \underline{x}_0 + K$$

só dat (12)

$$g_j(\underline{x}_0) = \underline{a}_j^* \underline{x}_0 - b_j^* \leq 0, \quad j \in W$$

waar

$$\underline{c}^* = (\underline{c}_0 - \underline{c}_a P^{-1} Q), \quad K = \underline{c}_a^* P^{-1} \underline{d} \\ \underline{a}_j^* = (\underline{a}_{j0} - \underline{c}_{ja} P^{-1} Q) \quad \text{en} \quad b_j^* = \underline{a}_{ja}^* P^{-1} \underline{d} - b_j;$$

en waar $\underline{c}_0$, $\underline{a}_{j0}$ en $\underline{c}_a$, $\underline{a}_{ja}$ onderskeidelik ooreenstem met die koëffisiënte van $\underline{x}_0$ en $\underline{x}_a$ in die oorspronklike doelfunksie en begrensings. Let op dat die gereduseerde LP-probleem van dimensie $n-p$ met $m-p$ begrensings is.

Stap 5. Bereken die volgende toelaatbare oplossing deur in die rigting $\nabla f^* = \underline{c}^*$ te stap:

$$\underline{x}_0^{i+1} = \underline{x}_0^i + h \underline{c}^* \\ \underline{x}_a^{i+1} = P^{-1} \underline{d} - P^{-1} Q \underline{x}_0^{i+1}$$

waar h soos vantevore [vergelykings (5)] bepaal word, maar dié keer m.b.t. Ω^* , die toelaatbare gebied van die gereduseerde probleem (12). Dit lewer dan ook 'n nuwe begrensing, sê g_s .

Stap 6. As $h=0$ en $v=0$, gaan na stap 7. As $h>0$ en $v=0$ of 1, stel dan $v \leftarrow 1$, stoor die ou aktiewe stel $W_{\text{oud}} \leftarrow W$, stel $p \leftarrow 1$, $w_1 \leftarrow s$ en gaan na stap 2. As $h=0$ en $v=1$ (huidige W net een element), dan as al die ou begrensings wat ooreenstem met W_{oud} aktief is (d.i. hulle berekende h 's is almal nul), voeg W_{oud} by W , $p \leftarrow p_{\text{oud}} + 1$, $v \leftarrow 0$ en gaan na stap 2; andersins stel $v \leftarrow 0$ en gaan na stap 7.

Stap 7. Stel $p \leftarrow p+1$ en $w_p \leftarrow s$.

Stap 8. As $p=n$ dan is $\underline{x}^{i+1}$ by 'n verteks en is dit 'n moontlike optimale oplossing; gaan dan na stap 9; andersins gaan na stap 2.

Stap 9. (Perturbasieprosedure) As $\underline{x}^{i+1}$ nie optimaal is nie, sal $f(\underline{x})$ toeneem langs minstens een van die $(n-1)$ -dimensionele rande wat wegloop vanuit die vertekspunt $\underline{x}^{i+1}$. Bereken nou geperturbeerde punte $\underline{y}^j, j=1,2,\dots,n$, wat elk op 'n afsonderlike rand lê deur die oplos van die stelsels:

$$P \underline{y}^j = \underline{d} - \underline{e}^j, \quad j=1,2,\dots,n$$

waar

$$\underline{e}^j = (0, 0, \dots, \underline{e}, \dots, 0)^t \\ \uparrow j\text{-de posisie}$$

en $\epsilon > 0$ (klein).

Die geperturbeerde punte word dus gegee deur:

$$\underline{y}^j = P^{-1} \underline{d} - P^{-1} \underline{e}^j \quad (13)$$

Toets nou $f(\underline{y}^j)$ teen $f(\underline{x}^{i+1})$, $j=1,2,\dots,n$. Indien $f(\underline{y}^j) \leq f(\underline{x}^{i+1})$, vir alle j dan is $\underline{x}^{i+1}$ optimaal en STOP (suksesvolle terminasie); andersins kies enige $\underline{y}^j$ (die eerste een, byvoorbeeld) waarvoor $f(\underline{y}^j) > f(\underline{x}^{i+1})$, stel $\underline{x}^1 \leftarrow \underline{y}^j$ en gaan na stap 1 (begin oor). Dit word hier aanvaar dat ϵ klein genoeg gekies word sodat $\underline{y}^j$ steeds 'n toelaatbare punt is.

3. IMPLEMENTERING VAN DIE METODE

3.1 Bepaling van toelaatbare beginpunt

Om die algoritme in die algemeen toe te pas vir die oplos van (1) moet 'n toelaatbare beginpunt altyd be-

skikbaar wees. In ons toepassing word die oorsprong altyd eerste vir 'n toelaatbare beginpunt getoets. Indien dit nie geskik is nie, word die sogenaamde hulp-probleem eers opgelos, nl.

$$\text{maksimeer } f(\underline{x}) = -x_{n+1}$$

só dat (14)

$$h_i(\underline{x}, x_{n+1}) = g_i(\underline{x}) - x_{n+1} \leq 0, \quad i=1,2,\dots,m. \\ h_{m+1}(x_{n+1}) = -x_{n+1} \leq 0.$$

Dit is duidelik dat $(0, a)$ 'n toelaatbare beginpunt van (14) is indien $a \geq -b_i$ vir $i=1,2,\dots,m$. Verder is dit ook duidelik dat die maksimum waarde van f nul is met $x_{n+1}=0$. Dit volg nou deur instelling dat indien ons 'n oplossing $(\underline{z}, 0)$ tot (14) kry, $\underline{z}$ 'n toelaatbare beginpunt vir ons oorspronklike probleem (1) is.

3.2 Substitusie van veranderlikes

In elke iterasie [vgl. (10) en (11) van stap 4] van die MP-algoritme word daar vir p afhanklike veranderlikes $\underline{x}_a$ opgelos uit die stelsel van p huidige aktiewe begrensings. 'n Prosedure word nou beskryf, waarvolgens hierdie oplossing, soos die aktiewe begrensings iterasie vir iterasie opbou, ekonomies bereken kan word. Die prosedure stem ooreen met die Gauss-Jordan-metode waar die oplossing in die volgende vorm verkry word:

$$I \underline{x}_a + P^{-1} Q \underline{x}_0 = P^{-1} \underline{d} \quad (15)$$

waar I die identiteitsmatriks is. Met die eerste iterasie is daar slegs een aktiewe begrensing ($p=1$). Sonder om verwarring te veroorsaak, dui ons die begrensing aan deur $\sum a_{1j} x_j = b_1$. In dié geval is die transformasie na (15) voor-die-hand-liggend. Vind 'n k sodat a_{1k} nie nul is nie. Daar moet so 'n k wees omdat die koëffisiënte van 'n begrensing nie almal nul kan wees nie. Die k -de komponent van $\underline{x}$ word nou $\underline{x}_a$ en die oorblywende komponente $\underline{x}_0$. Wanneer die vergelyking deur a_{1k} gedeel word, lewer dit die volgende:

$$x_k + (a_{1k})^{-1} \sum_{j \neq k} a_{1j} x_j = (a_{1k})^{-1} b_1$$

waar die sommasie oor alle i behalwe k gaan.

In die volgende iterasie sal daar of een of twee begrensings aktief wees. Indien een aktief is, word bostaande prosedure weer gevolg. Beskou nou die geval waar twee begrensings aktief is en aanvaar vir eenvoud dat, sonder verlies aan algemeenheid, die aktiewe begrensing van die vorige iterasie in die volgende vorm is:

$$x_1 + \sum a_{1j} x_j = b_1$$

Die ekstra nuwe begrensing is

$$a_{21} x_1 + \sum a_{2j} x_j = b_2$$

In matriksvorm kan die stelsel vergelykings nou soos volg voorgestel word:

$$\begin{bmatrix} 1 & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \end{bmatrix} \underline{x} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Pas nou Gauss-Jordan-eliminasië op die stelsel toe. Deur die gepaste veelvoud van die eerste vergelyking by die tweede te tel, word die eerste inskrywing van

die tweede vergelyking nul gemaak. Indien die tweede element op die tweede ry nul is, vind nou $k > 2$ sodat a_{2k} nie nul is nie. Indien die stelsel begrensings lineêr onafhanklik is, is ons verseker dat daar so 'n k bestaan. Ruil dan die tweede en k -de kolomme om. Vervolgens word die tweede vergelyking gedeel deur die tweede diagonaalelement. Laastens word die tweede inskrywing op die eerste ry nul gemaak deur die gepaste veelvoud van die tweede ry by die eerste te tel om dan die stelsel in die verlangde vorm (15) te gee.

Bostaande prosedure word nou herhaaldelik met die byvoeging van elke nuwe aktiewe begrensing per iterasie toegepas. Diagrammaties kan die algemene situasie, met die nuwe bykomende begrensing in die laaste ry, deur die volgende aangevulde matriks voorgestel word:

$$\left[\begin{array}{cccc|cccc|cccc} 1 & 0 & 0 & . & 0 & a & . & . & . & . & . & a & a & | & b \\ 0 & 1 & 0 & . & 0 & a & . & . & . & . & . & . & a & a & | & b \\ 0 & 0 & 1 & . & 0 & a & . & . & . & . & . & . & a & a & | & b \\ . & . & . & . & 0 & a & . & . & . & . & . & . & . & a & a & | & b \\ 0 & 0 & 0 & 0 & 1 & a & . & . & . & . & . & . & . & a & a & | & b \\ a & a & a & a & a & a & . & . & . & . & . & . & . & a & a & | & b \end{array} \right] \quad (16)$$

$p \times p$ $p \times (n-p)$ p -vektor

Meer formeel kan die algemene prosedure, wanneer die p -de begrensing bykom, in die volgende algoritmiese vorm gestel word:

Substitusiealgoritme

Stap 1. Doen vir $i = 1$ tot $p - 1$: maak a_{pi} nul deur die gepaste veelvoud ($-a_{pi}$) van die i -de ry by die p -de te tel.

Stap 2. As $a_{pp} = 0$ gaan na stap 5: andersins gaan voort.

Stap 3. Deel ry p deur a_{pp} .

Stap 4. Doen vir $i = 1$ tot $p - 1$: maak a_{ip} nul deur die gepaste veelvoud ($-a_{ip}$) van die p -de ry by die i -de ry te tel. Die verwerking van die p -de bykomstige begrensing is hiermee afgehandel met die afhanklike oplossing in die gevraagde vorm (15).

Stap 5. Vind 'n $k > p$ sodat a_{pk} nie nul is nie en ruil kolomme p en k [in stelsel (16)] om; gaan na stap 3. □

Indien die MP-algoritme termineer met n begrensings aktief, dan kan bostaande prosedure ook, as neweproduk, die nodige perturbasies, $-P^{-1}\epsilon^j$ (stap 9 van MP-algoritme), lewer wat nodig is vir die toets vir optimaliteit en die voortsetting van die oplossing. Dit word gedoen deur in bostaande eliminasieprocedure die matriks (16) aan die regterkant met vektore ϵ^j aan te vul.

4. EKSPERIMENTELE RESULTATE EN BESPREKING

Figuur 1(a) en (b) gee 'n driedimensionele meetkundige voorstelling van die werking van die nuwe algoritme wanneer dit toegepas word op onderskeidelik die eenvoudige probleem

(a) maksimeer $f = x_1 + 2x_2 + 3x_3$; sô dat $0 \leq x_i \leq 1$, $i = 1, 2, 3$

en die Klee-Minty-probleem

(b) maksimeer $f = 100x_1 + 10x_2 + x_3$ sô dat $x_1 \leq 1$,

$$20x_1 + x_2 \leq 100, \quad 200x_1 + 20x_2 + x_3 \leq 10\,000, \quad x_i \geq 0, \quad i = 1, 2, 3$$

Die figure illustreer duidelik hoe die MP-algoritme nie slegs stappe langs die eendimensionele rande van die oplossingspolitoopt neem nie, maar ook deur die interne ruimte en langs hoër-dimensionele vlakke beweeg. Vir probleem (b) vereis die simpleksmetode, in teenstelling met die 5 stappe van die MP-algoritme, dat 7 stappe langs eendimensionele rande geneem word, sodat al agt vertekse besoek word voordat die optimale punt bereik word.

Die MP-algoritme is in FORTRAN geprogrammeer en onder WATFIV op 'n HITACHI PS7/83 rekenaar getoets. Altesaam drie prototipe programme is ontwikkel, naamlik PROT1, PROT2 en

(a)

(b)

FIGUUR 1: Meetkundige voorstellings van die MP-stappe wat vereis word vir die oplos van probleme (a) en (b) soos aan die begin van paragraaf 4 gegee. Duidelikheidshalwe is figuur vir (b) nie volgens skaal geteken nie.

PROT3. Die aanvanklike program PROT1 het die IMSL lineêre oplosser LEQT1F¹¹ gebruik om die nodige perturbasies te doen wanneer die algoritme by 'n verteks termineer. PROT2 genereer outomaties die perturbasies soos beskryf aan die einde van paragraaf 3.2. PROT3 doen ook die perturbasies outomaties, maar pas ook spiering toe om moontlike onstabieleit a.g.v. afrondingsfoute uit te skakel. Spiering veroorsaak dat die baan voortgebring deur PROT3 gewoonlik verskillend van dié van die ander twee is, wat eksperimentele bewys is dat die baan nie uniek is nie, maar van die besondere substitusie-volgorde afhang. Die programme maak ook voorsiening vir die oplos van die hulpprobleem in die geval waar die oorsprong nie 'n toelaatbare beginpunt is nie. By die implementering van die algoritme op 'n rekenaar is dit belangrik om daarop te let dat a.g.v. die eindige woordlengte van die rekenaar die berekenings in die algemeen nie eksak is nie en afrondingsfoute voorkom. Dit is dus nodig om 'n toleransie (TOL) in te voer wat gebruik word wanneer ons bv. moet bepaal of 'n stapgrootte gelyk aan nul is en 'n begrensing gevolglik aktief mag wees. In dié geval bepaal ons of $|h| < \text{TOL}$, en indien wel, stel ons $h=0$. Tipies kies ons $\text{TOL} = 10^{-6}$. Verkiesliker, alhoewel nie hier geïmplementeer nie, behoort ons te toets vir $|h| * ||c^*|| < \text{TOL}$.)

Die programme is op tientalle probleme met $n \leq 20$ en $m \leq 60$ met sukses toegepas. Om te kontroleer dat die korrekte oplossings verkry is, is die probleme ook getoets aan die hand van die IMSL simpleks-subroetine ZX3LP.¹¹ Dit is nie prakties sinvol om al die probleme wat suksesvol opgelos is, hier te lys nie. Om egter 'n aanduiding te gee van die werking van die algoritme, gee ons die resultate vir enkele gesimuleerde willekeurige probleme van die vorm:

$$f(x) = \sum_{k=1}^n x_k$$

met begrensings

$$-x_k \leq 0 \quad k = 1, 2, \dots, n \quad (17)$$

en

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = 1, 2, \dots, m$$

waar a_{ij} willekeurige waardes (voortgebring deur die IMSL subroetine GGUBT¹¹ met aanvanklike SEED = 123457) tussen 0 en 1 000 aanneem met $b_i = 10\,000$ vir alle i . Die resultate word in tabel 1 gegee.

Vir klein waardes van n is die werking van die MP-algoritme redelik kompetend, maar vir baie groot n is die werking waarskynlik onprakties. Daar moet egter in gedagte gehou word dat die prototipe programme waarmee die algoritme hier getoets word, hoegenaamd nie met die oog op optimering van doeltreffendheid ontwerp is nie, maar slegs om aan te toon dat die algoritme met sukses toegepas kan word. Ook is dit belangrik om daarop te let dat toetsprobleme van die vorm (17) nie tipies van praktiese LP-probleme is nie. In groot LP-probleme wat in die praktyk voorkom, is die koëffisiëntematriks baie yl met 'n tipiese vulling van slegs 1%. Dit kan verwag word dat indien die programme aangepas word om in die Gauss-Jordan eliminasiëprosedure die ylheid van die koëffisiëntematriks uit te buit, die doeltreffendheid van die MP-algoritme in die geval van tipiese groot LP-probleme met ordes verbeter kan word.

Dit is ook beduidend dat die algoritme in baie gevalle die eerste keer termineer by òf die optimale verteks òf 'n naby-optimale verteks. Dit dui daarop dat die huidige algoritme, in ooreenstemming met die werk van Mitra et al.,⁶ in 'n hibriede skema gebruik kan word wat beide die nuwe metode en die simpleks-metode implementeer. Nie alleen kan die MP-algoritme 'n beter beginpunt vir die simpleksmetode voorsien nie, maar dit kan ook dien as 'n sogenaamde suiweringsprosedure waardeur 'n nie-basiese oplos-

TABEL 1

n	m	f_{opt}	PROT3		PROT2		ZX3LP	
			stappe*	tyd	stappe	tyd	iter.	tyd
5	5	20,28	4	0,11	4	0,10	2	0,04
5	10	16,12	4	0,15	4	0,13	4	0,11
5	15	14,53	4	0,16	4/1	0,28	6	0,24
8	8	13,86	7	0,57	7	0,53	7	0,14
8	16	13,80	8	0,99	7	0,81	9	0,40
8	40	13,72	8/1	2,71	8	1,56	11	1,85
10	10	19,62	11/1/1	2,39	11/1/1	2,44	9	0,20
10	40	16,58	10/1	2,94	9	2,83	9	1,69
12	24	17,59	12/2	5,51	14	5,46	16	1,18
15	15	19,28	17/1/2	12,29	18/2	12,10	15	0,58
20	20	19,67	20/2/1/1/1	42,48	26/1/2/1/1/1	53,89	30	1,75

*Die / na die aantal stappe dui aan dat die algoritme by 'n nie-optimale verteks termineer en dat na perturbasie verdere stappe soos aangedui, geneem word.

sing verbeter kan word na 'n ekstreempunt-oplossing, waarna die simpleksmetode toegepas kan word. Die uitvoering van hierdie suiweringsprosedure is belangrik waar iteratiewe interne soekmetodes, soos dié van Karmarkar,¹ gebruik word en waar dit gevind is dat 80% van die optimale waarde in ongeveer 25% van die iterasies verkry word. Die toepassing van 'n hibriede multipleks-simpleks prosedure in dié stadium sal moontlik die mees gepaste termineringstap vir sulke metodes wees.

BYLAE A: TEORETIESE EIENSKAPPE VAN DIE MP-ALGORITME

Enkele belangrike teoretiese eienskappe van die MP-algoritme word hieronder, sonder bewys, gegee.

Eienskap 1

As $\underline{c}^* = \underline{0}$ in enige stadium, dan, in die oorspronklike ruimte R^n , is $\underline{c}$ ortogonaal op die hipervlak gedefinieer deur die aktiewe begrensings en word die optimale oplossing bereik by elke toelaatbare punt op hierdie optimale hipervlak. $\square$

Eienskap 2

Die werklike staprigting in R^n is nie uniek nie, maar is afhanklik van die besondere substitusie; dit is, die keuse van afhanklike veranderlikes, waardeur die gereduseerde probleem verkry word. $\square$

In 'n rekenaarprogram sal hierdie substitusie, en dus die staprigting, in die algemeen afhang van die volgorde waarin die begrensings gespesifiseer is.

Eienskap 3

As die probleem 'n eindige aantal begrensings het en as die oplossing begrens is en $\underline{c}^* \neq \underline{0}$ vir alle stappe, dan sal die algoritme in 'n eindige aantal stappe by 'n verteks termineer, tensy dit 'n oneindige ry nie-nul stappe h_j neem, sodanig dat $\sum_{j=1}^{\infty} h_j$ konvergeer. $\square$

Die praktiese implikasie van hierdie eienskap is dat sou die oneindige ry nie-nul stappe teoreties voor-

kom, dan, omdat $h_j \rightarrow 0$ en ons met 'n toleransie TOL werk, daar 'n eindige k sal bestaan, sodat $h_k = 0$. Die gevolg hiervan is dat ons kan aantoon dat in die praktyk die algoritme onder bostaande voorwaardes altyd by 'n verteks sal termineer.

BEDANKING

Die outeurs bedank graag die Suid-Afrikaanse Wetenskaplike en Nywerheidsnavorsingsraad vir finansiële steun wat hierdie werk moontlik gemaak het.

Ontvang 20 Jan. 1987; aanvaar 20 Febr. 1987.

VERWYSINGS

1. Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming, *Combinatorica*, 4, 373-395.
2. Khachian, L.G. (1979). A polynomial algorithm in linear programming, *Soviet Mathematics Doklady* (translation), 20, 191-194.
3. Dantzig, G.B. (1963). *Linear Programming and Extensions* (Princeton University Press, New Jersey).
4. Todd, M.J. & Burrell, B.P. (1985). An extension of Karmarkar's algorithm for linear programming using dual variables, *Technical Report No. 648*, School of Operations Research and Industrial Engineering, Cornell University, New York.
5. Gill, P.E., Murray, W., Saunders, M.A., Tomlin, J.A. & Wright, M.H. (1985). On projected Newton barrier methods for linear programming and equivalence to Karmarkar's projective method, *Technical Report SOL 85-11*, Department of Operations Research, Stanford University, Stanford.
6. Mitra, G., Tamiz, M. & Yadegar, J. (1986). Experimental investigation of an interior search method within the simplex framework, *Technical Report TR/06/86*, Department of Mathematics and Statistics, Brunel University, Uxbridge.
7. Rosen, J.B. (1960). The gradient projection method for nonlinear programming: Part I, Linear constraints, *Journal of the Society for Industrial and Applied Mathematics*, 8, 181-217.
8. Wolfe, P. (1967). Methods of nonlinear programming, in Abadie, J. (Red.), *Nonlinear Programming*, 97-131, North Holland, Amsterdam.
9. Frisch, R. (1957). Linear dependencies and a mechanized form of the multiplex method for linear programming, Memorandum, Universitetets Sosialøkonomiske Institutt, Oslo.
10. Avis, D. & Chavatal, V. (1978). Notes on Bland's pivoting rule, *Mathematical Programming, Study 8*, 24-34.
11. International Mathematical and Statistical Libraries, IMSL Inc. Houston, Texas.