



# 'n Algoritme vir die minimum van die konkawe knapsakprobleem

G Venter\* JS Wolvaardt†

*Ontvang: 15 Januarie 1999; Hersien: 3 Desember 2004; Aanvaar: 27 Maart 2005*

## Abstract

This paper addresses the problem of resource allocation among activities where the cost of each is described by a concave function. There is a single linear constraint (limited resource) and each activity has an upper and lower bound (maximum and minimum resource allocations). The objective is to minimise the sum of the functions.

The problem with convex functions is well-studied and since a local minimum is also global, this problem was tamed early by Luss and Gupta [LUSS H & GUPTA SK, 1975, *Allocation of effort resources among competing activities*, Operations Research, **23**, pp. 360–366], and by Bitran and Hax [BITRAN GR & HAX AC, 1979, *On the solution of convex knapsack problems with bounded variables*, pp. 357–367 in PRÉKOPA A (ED.), *Survey of Mathematical Programming*]. In contrast the minimisation of a sum of concave functions has received less attention and then the emphasis has often been on an objective function which is nonseparable quadratic, and on the complexity of finding a true local minimiser (*e.g.* Moré and Vavasis [MORÉ JJ & VAVASIS SS, 1991, *On the solution of concave knapsack problems*, Mathematical Programming, **49**, pp. 397–411]).

We are concerned with the computational problem of finding the global optimum for a (separable) sum of nondecreasing general concave functions and the approach is via the Kuhn-Tucker necessary conditions. These are improved by using the result that a minimiser must be an extreme point, which means that all but one variable (at most) is at an upper or lower bound. (Moré and Vavasis base their CKP algorithm on this property.) The improved necessary conditions form the basis of the method of greatest differences (GVA), our algorithm to improve a feasible solution. A greedy heuristic to produce a first feasible solution is also proposed.

Using four groups of 10 instances, each from the four classes of concave functions of Luss and Gupta, one thousand different runs for incremental values of the resource were made for both the CKP and our GVA. While the CKP often found a globally suboptimal answer for functions which intersect, the GVA found the correct answer in all 16 000 runs. This is no guarantee that a method based on necessary conditions will always find the global maximum but the GVA is numerically promising and it masters a class of problems that the CKP does not. The greedy first feasible solution was found to be optimal more frequently (64% to 49%, and 73% to 40%) than that proposed by Moré and Vavasis.

The GVA does not depend on the kind of function, requiring only that the functions be nondecreasing and concave. With minor alterations it can be used for the maximisation of the sum of nondecreasing convex functions. It is much faster than dynamic programming on problems with up to 10 functions and should be superior when tested on large problems.

---

\*Korresponderende outeur: Instituut vir Maritieme Tegnologie, Posbus 181, Simonstad, 7995, Suid-Afrika, e-pos: [gve@imt.co.za](mailto:gve@imt.co.za)

†Departement Besluitkunde, Universiteit van Suid-Afrika, Posbus 392, Pretoria, 0003, Suid-Afrika

**Key words:** Resource allocation problem, nonlinear knapsack problem, nonconvex optimisation.

### Opsomming

Hierdie artikel beskrywe die hulpbrontoekeningsprobleem van aktiwiteite waar die koste van elkeen gegee word deur 'n konkawe funksie. Daar is 'n enkele lineêre beperking en elke aktiwiteit het 'n bo- en ondergrens. Die doelwit is die minimering van die som van die kostefunksies.

Die probleem waar die doelfunksies konveks is, is baie goed bestudeer. In teenstelling hiermee het die minimering van die som van konkawe doelfunksies baie minder aandag in die literatuur gekry en die klem was dikwels op 'n doelfunksie wat nieskeibaar en kwadratiese is, en op die kompleksiteit om die plaaslike minimum te verkry. Die enigste algoritme wat hierdie probleem oplos is die CVA-algoritme van Moré en Vavasis [MORÉ JJ & VAVASIS SS, 1991, *On the solution of concave knapsack problems*, Mathematical Programming, **49**, pp. 397–411].

Die probleem om die globale minimum vir 'n som van niedalende algemene konkawe funksies te verkry, word hier aangepak. Die benadering is via die Kuhn-Tucker nodige voorwaardes. Hierdie metode is verbeter deur die resultaat te gebruik dat by die algehele minimum hoogstens een funksie nie by sy onder- of bogrens is nie. Hierdie verbeterde nodige voorwaarde vorm die basis van die nuwe algoritme wat die uitvoerbare oplossing verbeter, naamlik die grootste verskillende algoritme (GVA).

Deur gebruik te maak van 10 voorkomste elk van die vier klasse van konkawe funksies van Luss en Gupta [LUSS H & GUPTA SK, 1975, *Allocation of effort resources among competing activities*, Operations Research, **23**, pp. 360–366], is een duisend verskillende lopies met inkrementele waardes vir die hulpbron gedoen met die CKP en GVA algoritmes. Terwyl die CKP verskeie kere 'n globale suboptimale antwoord vir funksies wat mekaar sny gekry het, het die GVA die korrekte optimale antwoord in al 16 000 lopies verkry. Dit is egter nie 'n waarborg dat 'n metode wat gebaseer is op die nodige voorwaardes altyd die globale optimum sal verkry nie, maar die GVA is numeries baie belowend en dit spreek 'n klas van probleme aan wat die CKP metode nie doen nie.

Die GVA is nie afhanklik van die tipe funksies nie — daar word net vereis dat die funksies niedalend en konkav moet wees. Met klein aanpassings kan die metode ook gebruik word vir die maksimering van die som van niedalende konvekse funksies.

**Sleutelwoorde:** Hulpbrontoekeningsprobleem, nielineêre knapsakprobleem, niekonvekse optimering.

## 1 Inleiding

In hierdie artikel word daar ondersoek ingestel na 'n spesiale geval van die hulpbrontoekeningsprobleem. In die praktyk kom dié probleem voor by die minimering van koste onderhewig aan skaalvoordele. Voorbeelde hiervan word gevind by produksiebeplanning [3, 7], kapasiteitsverbreding [14, 17], ingenieursontwerp [11], en netwerkvloeiopname [4, 5].

Vir die outeurs het die probleem 'n praktiese oorsprong in 'n meer algemene probleem waar 'n beperkte hulpbron,  $B$ , tussen 'n aantal aktiwiteite verdeel moet word om die totale opbrengs te maksimeer. Aktiwiteit  $i$  kry 'n hoeveelheid  $x_i$  van die hulpbron en lewer 'n opbrengs  $f_i(x_i)$ . Elke aktiwiteit het sy eie produksiefunksie  $f_i$  en elke funksie  $f_i$  behoort tot een van drie klasse: konkav, konveks of S-vormig. Al die funksies  $f_i$  is niedalend.

Ons praktiese probleem is ongewoon daarin dat al drie klasse funksies gelyktydig kan voorkom. (Lootsma, *et al.* (1990) gebruik wel al drie in 'n uitvoerige sensitiviteitsanalise,

maar dan om die beurt.) Dit is te begrype dat net een klas kan voorkom in gevalle waar produksiefunksies dalende meeropbrengs toon (konkaaf) soos die toewysing van be- markingsfondse van Luss en Gupta (1975). Die maksimering van opbrengs het dikwels 'n ekwivalent in die minimering van koste en hier is die ekwivalente probleem die minime- ring van 'n som van kostefunksies wat stygende marginale koste toon (konveks) soos die kragopwekkingskeduleringsprobleem van Van den Bosch en Lootsma (1987).

Die hulpbrontoekenningsprobleem is 'n veralgemening van die bekende heeltallige knap- sakprobleem met dié verskil dat die heeltallige knapsakprobleem 'n lineêre doelfunksie het wat die nut voorstel van 'n aantal items waarvan daar een van elke soort óf ingesluit óf uitgesluit moet word. Hier het ons egter aktiwiteite waarvan meer of minder gedoen kan word (binne die bo- en ondergrense), die hulpbron is ook beperk, en die doelfunksie is die som van die produksiefunksies. Die probleem word as volg gedefinieer:

$$\left. \begin{array}{ll} \text{Maksimeer} & \sum_{i=1}^n f_i(x_i) & \text{(totale opbrengs)} \\ \text{onderhewig aan} & \sum_{i=1}^n x_i = B & \text{(beperkte hulpbron)} \\ & a_i \leq x_i \leq b_i, \quad i = 1, 2, \dots, n & \text{(bo- en ondergrense)} \end{array} \right\} \quad (1)$$

waar alle funksies  $f_i(x_i)$  nedalend en óf konveks, óf konkaaf, óf S-vormig is, en alle  $a_i \geq 0$ , vir  $i = 1, 2, \dots, n$ .

Dié probleem kom in die praktyk voor by die opstelling van nulbasisbegrotings. Produk- siefunksies van al drie soorte kan in een model voorkom, en dit het al gebeur, maar die produksiefunksies is meestal 'n mengsel van konkaaf en konveks. Alhoewel die probleem heeltemal bevredigend met dinamiese programmering opgelos kan word, sou dit weten- skaplik meer bevredigend wees om 'n metode te gebruik wat die struktuur en eienskappe van die probleem uitbuit.

Ons aanslag was om eers na die geval waar al die funksies  $f_i$  konkaaf is (sien die volgende afdeling) te kyk. Hierdie probleem kan as opgelos beskou word. Die volgende stap was 'n algoritme vir die maksimering van die som van konvekse produksiefunksies wat in hierdie artikel aangebied word as die minimering van die som van konkawe funksies. Die maksi- mering van die som van 'n mengsel van konkawe, konvekse en S-vormige produksiefunksies is die uiteindelijke doelwit.

Omdat produksiefunksies met dalende meeropbrengs so dikwels voorkom is dit nie 'n verrassing nie dat die meeste werk gedoen is op die probleem waar al die funksies  $f_i$  konkaaf is. 'n Kort literatuuroorsig word in die volgende afdeling gegee.

## 2 Konkawe maksimering

Charnes en Cooper (1958) het in 'n vroeë artikel oor optimale soeking die totale waarskyn- likheid dat 'n teiken gevind sal word, beskryf as die som van die waarskynlikheidsfunksies van die verskillende pogings  $f_i(x_i) = s_i [1 - \exp(-mx_i)]$ ,  $i = 1, 2, \dots, n$ . Hulle het kon- vekse programmering ingespan.

Wilkinson en Gupta (1969) het 'n ander toepassing beskou, maar hulle funksies het die ietwat meer algemene vorm  $f_i(x_i) = s_i [1 - \exp(-m_i x_i)]$ ,  $i = 1, 2, \dots, n$ . Alhoewel hulle die Kuhn-Tuckervoorwaardes noem, gebruik hulle dinamiese programmering.

Luss en Gupta (1975) gee die beginsels van 'n iteratiewe algoritme gebaseer op die Kuhn-Tuckervoorwaardes en die grensverslappingsprosedure (net die bogrense). Vir vier tipes konkawe funksies (sien (2) in die volgende afdeling) spesialiseer hulle dan die prosedure om 'n enkelgang algoritme daar te stel. (In elke stap van die algoritme word minstens één veranderlike by sy optimumwaarde vasgestel.)

Bitran en Hax (1979) het onder andere die vier versamelings konkawe funksies van Luss en Gupta gebruik in die toetsing van hulle algoritme — die berekeningspoed is heelwat vinniger as dié van Luss en Gupta. Hulle het die onder- sowel as die bogrensbepelings verslap. Ook hier word daar by elke stap minstens een veranderlike op een van sy grense vasgepen.

Ibaraki en Katoh (1988) het vir hulle “skeibare konvekse minimeringsprobleem” dieselfde vier versamelings funksies as Luss en Gupta gebruik. Soos by Luss en Gupta word die funksies aanvanklik gerangskik volgens die grootte van die afgeleides  $f'_i(a_i)$ ,  $i = 1, 2, \dots, n$ . Die algoritme is nou verwant aan dié van Luss en Gupta. Hulle rapporteer verskeie praktyktoepassings van hulle algoritme.

Van den Bosch en Lootsma (1987) het 'n skeduleringsprobleem by kragopwekking gemodelleer met kwadratiese kostefunksies maar ook met 'n enkele nieskeibare nielineêre kostefunksie. Vir die eerste model het hulle dieselfde algoritme gebruik as Bitran en Hax (1979). Vir die tweede model gebruik (en vergelyk) hulle beide die gradiëntprojeksiemethode en die gereduseerde-gradiëntmethode.

Die Kuhn-Tuckervoorwaardes word as vertrekpunt gebruik wanneer ons die maksimeringsprobleem ondersoek waar die produksiefunksies almal konveks is; en dit maak dus sin om eers na Luss en Gupta (1975) se resultate vir die konkawe maksimeringsprobleem te kyk wat ook met behulp van die Kuhn-Tuckervoorwaardes aangepak is.

### 3 Die werk van Luss en Gupta

Luss en Gupta (1975) het drie teoretiese resultate ontwikkel en toegepas.

1. Beskou probleem (1) as 'n konkawe maksimeringsprobleem. Nodige en voldoende voorwaardes vir 'n uitvoerbare oplossing  $\underline{x} = \underline{x}^*$  om optimaal te wees, is dat daar 'n nienegatiewe getal  $M$  bestaan (die Lagrangevermenigvuldiger van die begrotingsbeperking) sodanig dat  $\sum_{i=1}^n x_i = B$ : as  $x_i = a_i$  dan is

$$\frac{d}{dx} f_i(a_i) \leq M, \quad i \in I = \{i \mid x_i = a_i\},$$

as  $a_k < x_k < b_k$  dan is

$$\frac{d}{dx} f_k(x_k) = M, \quad k \in K = \{k \mid x_k \in (a_k, b_k)\}$$

en as  $x_j = b_j$  dan is

$$\frac{d}{dx} f_j(b_j) \geq M, \quad j \in J = \{j \mid x_j = b_j\}.$$

2. As  $\underline{y}^* = (y_1^*, y_2^*, \dots, y_n^*)$  die optimale oplossing is met die boonste grense  $b_i$  weggelaat vir alle waardes van  $i$ , en as  $y_i^* \geq b_i$ , dan volg dat  $x_i^* = b_i$  in die oorspronklike model, vir alle  $i = 1, 2, \dots, n$ .
3. As  $\frac{d}{dx} f_i(a_i) \geq \frac{d}{dx} f_{i+1}(a_{i+1})$  vir  $i = 1, 2, \dots, n - 1$  dan bestaan daar 'n heelgetal  $p$ , met  $1 \leq p \leq n$ , sodanig dat  $x_i > a_i$ ,  $i = 1, 2, \dots, p$  en  $x_i = a_i$ ,  $i = p + 1, p + 2, \dots, n$ .

'n Algoritme gebaseer op hierdie resultate is ontwikkel en toegepas op die vier klasse konkawe funksies

$$\left. \begin{aligned} f_i(x_i) &= s_i(1 - e^{-m_i x_i}), \\ f_i(x_i) &= s_i x_i - m_i x_i^2, \\ f_i(x_i) &= s_i(x_i + c_i)/(x_i + m_i), \quad m_i > c_i \\ \text{en } f_i(x_i) &= s_i \ln(1 + m_i x_i). \end{aligned} \right\} \quad (2)$$

Die waardes van die funksies  $M(p)$  (die Lagrangevermenigvuldiger  $M$  uitgedruk as 'n funksie van  $p$ ) en  $\frac{d}{dx} f_{p+1}(a_{p+1})$  word vir elke funksie bereken en in hulle algoritme gebruik om die optimale  $x_i^*$ -waardes,  $i = 1, 2, \dots, n$ , in 'n enkelgang te bereken. Die algoritme maak gebruik van die differensieerbaarheid van die funksies en van die vorm van die funksie. Ander differensieerbare funksies kan in beginsel net so hanteer word deur die funksie se eie  $M(p)$ -waarde vooraf te bereken en dan hulle algoritme toe te pas.

## 4 Konkawe minimering: Die werk van Moré en Vavasis

Geen verwysings kon gevind word na die maksimering van konvekse funksies nie. Die ekwivalente probleem vir die minimering van konkawe funksies word wel gevind in 'n artikel van Moré en Vavasis (1991) wat beweer dat voor hulle publikasie die “streng konkawe geval nie aandag gekry het nie.” Hulle takel dié probleem deur by streng plaaslike minimeerders te begin. Hulle doelwit is om die kompleksiteit van dié probleem te verlaag en hulle slaag ook daarin. Hulle CKP (Eng.: *Concave Knapsack Problem*) algoritme los die probleem op in polinomialbegrensde tyd met goeie grense vir die geval waar die funksies  $f_i(x_i)$  kwadratiese is; 'n probleem wat self NP-moeilik is.

Om direkte vergelyking met hulle werk moontlik te maak, word met hulle formulering gewerk. Die ekwivalente probleem om die minimum van die som van konkawe funksies te kry, word die onderwerp. Dit staan ook bekend as die konkawe (minimerende) knap-

sakprobleem of kortweg die konkawe minimeringsprobleem, naamlik

$$\left. \begin{array}{l} \text{Minimeer} \quad \sum_{i=1}^n f_i(x_i) \\ \text{onderhewig aan} \\ \sum_{i=1}^n x_i = B \\ a_i \leq x_i \leq b_i, \quad i = 1, 2, \dots, n, \end{array} \right\} \quad (3)$$

waar alle funksies  $f_i(x_i)$ ,  $i = 1, 2, \dots, n$ , nedalend en konkaf is.

Aanvaar, sonder verlies aan algemeenheid, dat  $a_i < b_i$  vir alle  $i = 1, 2, \dots, n$  (as  $b_i = a_i$ , kan  $x_i$  deur 'n konstante vervang word en wel  $x_i = a_i = b_i$ ). Aanvaar verder dat  $\sum_{i=1}^n a_i < B < \sum_{i=1}^n b_i$ . (As  $B$  nie hierdie beperking bevredig nie geld óf  $B < \sum_{i=1}^n a_i$  en (3) is onuitvoerbaar, óf  $B = \sum_{i=1}^n a_i$  óf  $B = \sum_{i=1}^n b_i$  en daar is slegs een uitvoerbare punt óf  $B > \sum_{i=1}^n b_i$  en die hulpbron is nie beperk nie en daar is nie 'n toekenningsprobleem nie.)

Moré en Vavasis (1991) se CKP-algoritme vir die plaaslike minimum van (3) is gebaseer op 'n karakterisering van die plaaslike minimum wat hulle met uitvoerbare rigtings aflei. Nogtans gebruik hulle by implikasie (anders afgelei en anders geformuleer) die resultate wat ons in die volgende afdeling as Stellings 1 en 2 formuleer. Hulle motiveer die vind van 'n plaaslike minimum deur te argumenteer dat dit tog 'n bogrens op die algehele minimum plaas en dikwels die algehele minimum sal bereik. En dit is presies wat gebeur.

Numeriese werk toon dat die CKP-algoritme wel die minimum vind vir 'n versameling nedalende (oor die gebied wat beskou word) kwadratiese funksies uit die oorsprong (versameling 1) wat mekaar nie sny nie. 'n Tweede versameling wat dieselfde eienskappe het, is ook beskou (versameling 2) maar só saamgestel dat sommige van hulle mekaar sny. Die CKP-algoritme vind die minima vir (3) as die funksies uit versameling 1 is, maar as versameling 2 se funksies gebruik word, vind die CKP-algoritme dikwels suboptimale plaaslike minima.

Indien daar snydings van funksies voorkom, kan die CKP-algoritme dus nie gebruik word vir die berekening van die algehele minimum van die konkawe knapsakprobleem nie. Die eerste taak is om 'n algoritme te vind wat minstens hierdie klas funksies kan hanteer.

## 5 Nodige voorwaardes

Herskryf (3) vir die afleiding van die Kuhn-Tuckervoorwaardes (Luenberger (1984)):

$$\left. \begin{array}{l} \text{Minimeer} \quad \sum_{i=1}^n f_i(x_i) \quad \text{(totale opbrengs)} \\ \text{onderhewig aan} \\ \sum_{i=1}^n x_i - B = 0 \quad \text{(hulpbron)} \quad (\alpha) \\ -x_i + a_i \leq 0, \quad i = 1, 2, \dots, n \quad \text{(bogrens)} \quad (\beta_i) \\ x_i - b_i \leq 0, \quad i = 1, 2, \dots, n, \quad \text{(ondergrens)} \quad (\gamma_i) \end{array} \right\} \quad (4)$$

waar alle funksies  $f_i(x_i)$ ,  $i = 1, 2, \dots, n$ , differensieerbaar is. Die Lagrangefunksie vir (4) is

$$L(x_i, \alpha, \beta_i, \gamma_i) = \sum_{i=1}^n f_i(x_i) + \alpha \left( \sum_{i=1}^n x_i - B \right) + \sum_{i=1}^n \beta_i (-x_i + a_i) + \sum_{i=1}^n \gamma_i (x_i - b_i).$$

Volgens die Kuhn-Tuckervoorwaardes geld vir elke  $i$  by optimaliteit dat

$$\left. \begin{aligned} f'_i(x_i) + \alpha - \beta_i + \gamma_i &= 0, \\ \beta_i(-x_i + a_i) &= 0, \\ \gamma_i(x_i - b_i) &= 0, \\ \beta_i &\geq 0 \\ \text{en } \gamma_i &\geq 0. \end{aligned} \right\} \quad (5)$$

Laat  $\underline{x} = \underline{x}^*$  'n optimale uitvoerbare oplossing wees. Definieer die indeksversamelings  $I = \{i \mid x_i^* = a_i\}$ ,  $J = \{j \mid x_j^* = b_j\}$  en  $K = \{k \mid x_k^* \in (a_k, b_k)\}$ . Die Kuhn-Tuckervoorwaardes in (5) spesialiseer na drie gevalle. Vir alle  $i \in I$  geld dat

$$\beta_i \geq 0, \gamma_i = 0 \text{ en } f'_i(a_i) + \alpha - \beta_i = 0, \quad (6)$$

vir alle  $j \in J$  dat

$$\beta_j = 0, \gamma_j \geq 0 \text{ en } f'_j(b_j) + \alpha + \gamma_j = 0 \quad (7)$$

en vir alle  $k \in K$  dat

$$\beta_k = 0, \gamma_k = 0, f'_k(x_k) + \alpha = 0, \quad (8)$$

waaruit volg dat  $\alpha = -f'_k(x_k)$  vir alle  $k \in K$ . Deur  $\alpha$  in (6) te stel en omdat in (6) alle  $\beta_i \geq 0$  is, volg dat

$$f'_i(a_i) \geq f'_k(x_k), \quad i \in I \text{ en } k \in K.$$

Deur  $\alpha$  in (7) te stel waar alle  $\gamma_i \geq 0$  is, volg netso dat

$$f'_k(x_k) \geq f'_j(b_j), \quad j \in J \text{ en } k \in K.$$

Hierdie afleiding kan formeel saamgevat word in die volgende stelling wat (vir die maksimeringsprobleem) aan sowel Ibaraki en Katoh (1988) as Luss en Gupta (1975) bekend is. (Alhoewel Moré en Vavasis (1991) die probleem met uitvoerbare rigtings aanpak, gebruik hulle by implikasie ook dié stelling.)

### Stelling 1 (Kuhn-Tucker nodige voorwaardes)

Beskou probleem (4) en laat  $\underline{x} = \underline{x}^*$  'n optimale oplossing wees, dan bestaan daar 'n reële  $\alpha$  waarvoor geld dat

$$\left. \begin{aligned} f'_i(x_i^*) &\geq -\alpha, & i \in I, \\ f'_j(x_j^*) &\leq -\alpha, & j \in J \\ \text{en } f'_k(x_k^*) &= -\alpha, & k \in K, \end{aligned} \right\} \quad (9)$$

waar  $f_i(x_i)$ ,  $i = 1, 2, \dots, n$ , differensieerbare funksies is. ■

Indien die funksies  $f_i(x_i)$ ,  $i = 1, 2, \dots, n$ , stygend konkaf is, volg uit (9) dat  $\alpha \leq 0$ .

**Stelling 2**

Beskou probleem (4) (waar die funksies  $f_i(x_i)$ ,  $i = 1, 2, \dots, n$ , konkawe funksies is). As  $\underline{x} = \underline{x}^*$  'n optimale oplossing is, dan bestaan daar hoogstens een funksie  $k \in K$  waarvoor geld dat  $x_k^* \in (a_k, b_k)$ .

**Bewys:** Omdat die funksies  $f_i(x_i)$ ,  $i = 1, 2, \dots, n$  van probleem (3) konkav is, is hulle som ook konkav. As  $\underline{x}^*$  'n optimale oplossing vir (3) is, minimeer dit dus 'n konkawe funksie oor 'n geslote konvekse gebied, wat beteken (Luenberger (1984), p.181) dat  $\underline{x}^*$  'n ekstreme punt is. Definieer nou die hipervlakke  $A_i = \{\underline{x} \mid x_i = a_i\}$  en  $B_i = \{\underline{x} \mid x_i = b_i\}$  vir  $i = 1, 2, \dots, n$  en laat  $H = \{\underline{x} \mid \sum_{i=1}^n x_i = B\}$ . Elke ekstreme punt van (3) se uitvoerbare gebied is die snyding van  $H$  en  $n - 1$  van die hipervlakke  $A_i$  en  $B_i$ , waar  $A_s$  en  $B_s$  (met dieselfde indeks  $s$ ) nie saam voorkom nie. Omdat  $\underline{x}^*$  'n ekstreme punt is, geld dit vir hom ook, dit wil sê  $n - 1$  van sy  $n$  elemente is óf by 'n ondergrens óf by 'n bogrens. Dit laat hoogstens een element  $x_k$  waarvoor kan geld dat  $x_k^* \in (a_k, b_k)$ . ■

Die vorige twee resultate kan in die volgende stelling saamgevoeg word.

**Stelling 3 (Kuhn-Tucker nodige voorwaardes vir konkawe optimaliteit)**

Vir probleem (3) waar alle  $f_i(x_i)$ ,  $i = 1, 2, \dots, n$ , differensieerbare konkawe funksies is, geld dat as  $\underline{x} = \underline{x}^*$  'n optimale oplossing is, dan bestaan daar hoogstens een funksie  $f_k(x_k)$ , met  $x_k \in (a_k, b_k)$ , sodanig dat  $f'_k(x_k^*) = -\alpha$ ,  $f'_i(x_i^*) \geq -\alpha$ , vir alle  $i \in I$  en  $f'_j(x_j^*) \leq -\alpha$ , vir alle  $j \in J$ . ■

Uit Stelling 2 volg dat as  $\underline{x} = \underline{x}^*$  'n optimale oplossing is, dit ekwivalent voorgestel kan word deur die kategoriese stelsel  $\{I, J, \{k\}\}$  wat uit drie indeksversamelings bestaan en waar  $I = \{i \mid x_i^* = a_i\}$ ,  $J = \{j \mid x_j^* = b_j\}$ , en met  $k$  die indeks van die veranderlike waarvoor kan geld dat  $x_k^* \in (a_k, b_k)$ .

Ons verwys na 'n uitvoerbare oplossing van hierdie soort as 'n bo-onder-een-tussen-oplossing, kortweg 'n *boetoplossing*. Die veranderlikes wat met  $I$  geïndekseer is, word *onaktief* genoem, en die res *aktief*.

## 6 Die grootste-verskilalgoritme

Die nodige voorwaardes wat hierbo afgelei is, kom baie ooreen met dié wat Luss en Gupta (1975) (Afdeling 3) vir die konkawe maksimeringsprobleem verkry het, maar in ons geval is daar hoogstens een veranderlike wat nie op óf sy ondergrens óf sy bogrens moet wees nie. Dit laat die probleem baie maklik lyk — veral as die hellings in Stelling 3 ook nog in gedagte geneem word.

Die eerste algoritme wat 'n mens te binne skiet is om die funksies te rangskik volgens hulle hellings by hulle ondergrense en van groot na klein in  $I$  te plaas met een oog op die hulpbron totdat dit tyd word vir  $x_k$ . Die res is in  $J$ . (Die probleem kan natuurlik gereduseer word deur te sê dat alle veranderlikes op hulle laagste vlak in  $J$  is, die hulpbron  $B$  te verminder met  $\sum a_i$ , en die funksies te laat loop van 0 na  $b_i - a_i$ ). Omgekeerd kan 'n mens bo begin en volgens die helling in  $b_k$  vir  $J$  opbou totdat die hulpbron sê dat dit



nou tyd vir  $x_k$  is. Dit is dan ook die manier wat Moré en Vavasis (1991) gebruik om hulle beginoplossing te vind.

As die funksies lineêr was, was die probleem triviaal, want dan het 'n mens  $J$  se elemente ingepak volgens die beginsel van die minste doelfunksiewaarde per eenheid van die hulpbron en die laaste bietjie volgemaak uit die volgende laagste een. (Dis trouens hoe die meeste van ons dink wanneer 'n mens vir die eerste keer na die gewone heeltallige knapsakprobleem kyk!) Omdat die funksies konkaf is, kan die gemiddelde doelfunksiewaarde per eenheid van die hulpbron by  $x_k$  egter hoër wees as een van dié wat alreeds ewe gulsig ingesit is. Dit bring 'n mens terug by die dilemma van die heeltallige knapsakprobleem; dit lyk eenvoudig maar is dit hoegenaamd nie.

Alhoewel 'n gulsige (Eng.: *greedy*) oplossing nie noodwendig optimaal is nie, kan hy dit in eenvoudige gevalle wel wees en in die ander gevalle lê hy hopelik na aan die werklike optimum. Ons gebruik dan ook die gulsige keusemetode van die vorige paragraaf as heuristiese metode om 'n beginoplossing daar te stel. Die sukses van 'n beginoplossing word veral gemeet aan die persentasie van die kere wat dit optimaal is en in die volgende afdeling word 'n numeriese vergelyking met dié van Moré en Vavasis (1991) gerapporteer.

Beginoplossings impliseer algoritmes wat verder soek totdat optimaliteit gevind is. Die skryfwyse  $\{I, J, \{k\}\}$  vir 'n boetoplossing herinner aan lineêre programmering se basiese oplossings en die nieaktiewe veranderlikes stem met lineêre programmering se niebasiese veranderlikes ooreen. Die nodige voorwaardes sê dan ook (soos in lineêre programmering met basiese oplossings) dat 'n mens net na boetoplossings hoef te kyk, want 'n optimale oplossing is noodwendig 'n boetoplossing. Die taak is nou om 'n algoritme te kry om van 'n gegewe boetoplossing na 'n beter een te beweeg. Die algoritme gebaseer op Stelling 4 hieronder oorweeg die nieaktiewe veranderlikes om hulle aktief te maak en doen dit as dit die doelfunksie se waarde verlaag.

Dis die algoritmiese siening. Analities sou 'n mens verwag dat 'n beter oplossing as 'n boetoplossing net moontlik is as die nodige voorwaardes van Stelling 3 nie voldoende is nie (dit is maklik om met 'n teenvoorbeeld aan te toon dat Moré en Vavasis (1991) se beginoplossings boetoplossings is, waarvan 'n groot persentasie nie optimaal is nie) en sterker nodige voorwaardes moet dan gevind word. Stelling 4 gee sterker nodige voorwaardes.

**Stelling 4 (Sterker nodige voorwaardes vir optimaliteit: 'n Beter oplossing)**

Beskou probleem (3), waar alle funksies  $f_i(x_i)$  nedalend en konkaf is, met alle funksies  $f_i(a_i) \geq 0$ ,  $i = 1, 2, \dots, n$ . Laat  $\underline{x}$ , oftewel  $\{I, J, \{k\}\}$ , 'n uitvoerbare oplossing wees wat aan die nodige voorwaardes van Stelling 3 voldoen.

Kies vir elke  $p \in I + \{k\}$  'n  $Q_p \subset J$  en  $r_p \in J$ ,  $r \notin Q_p$  sodanig dat

$$\begin{aligned} \sum_{q \in Q_p} [f_q(b_q) - f_q(a_q)] + f_{r_p}(b_{r_p}) - f_{r_p}(x_{r_p}) \\ = \max_{j \in J, Q \subset J, j \notin Q} \sum_{i \in Q} [f_i(b_i) - f_i(a_i)] + f_j(b_j) - f_j(x_j), \end{aligned} \quad (10)$$

waar  $x_j = \sum_{q \in Q} [b_q - a_q] + b_j - [b_p - a_p] + [x_k - a_k]$  en  $a_j \leq x_j \leq b_j$ .

Die nodige voorwaardes vir 'n uitvoerbare oplossing  $\underline{x}^*$  om optimaal te wees, is dan dat

$$\sum_{q \in Q_p} [f_q(b_q) - f_q(a_q)] + f_{r_p}(b_{r_p}) - f_{r_p}(x_{r_p}) \leq [f_p(b_p) - f_p(a_p)] - [f_k(x_k) - f_k(a_k)] \quad (11)$$

vir alle  $p \in I + \{k\}$ .

**Bewys:** Veronderstel vir 'n oplossing  $\underline{x}$  dat (11) nie geld nie, met ander woorde daar bestaan 'n  $p$  waarvoor

$$\sum_{q \in Q_p} [f_q(b_q) - f_q(a_q)] + f_{r_p}(b_{r_p}) - f_{r_p}(x_{r_p}) > [f_p(b_p) - f_p(a_p)] - [f_k(x_k) - f_k(a_k)]. \quad (12)$$

Die totale opbrengs van die optimale oplossing is

$$\sum_{j \in J} f_j(b_j) + \sum_{i \in I} f_i(a_i) + f_k(x_k).$$

Tel hierdie totale opbrengs by aan beide kante van (12) om die ongelykheid

$$\begin{aligned} & \sum_{j \in J} f_j(b_j) + \sum_{i \in I} f_i(a_i) + f_k(x_k) + \sum_{q \in Q_p} [f_q(b_q) - f_q(a_q)] + f_{r_p}(b_{r_p}) - f_{r_p}(x_{r_p}) \\ & > \sum_{j \in J} f_j(b_j) + \sum_{i \in I} f_i(a_i) + f_k(x_k) + [f_p(b_p) - f_p(a_p)] - [f_k(x_k) - f_k(a_k)] \end{aligned}$$

te verkry. Dit reduceer na

$$\begin{aligned} & \sum_{j \in J} f_j(b_j) + \sum_{i \in I} f_i(a_i) + f_k(x_k) + \sum_{q \in Q_p} [f_q(b_q) - f_q(a_q)] + f_{r_p}(b_{r_p}) - f_{r_p}(x_{r_p}) \\ & > \sum_{j \in J} f_j(b_j) + \sum_{i \in I} f_i(a_i) + [f_p(b_p) - f_p(a_p)] + f_k(a_k). \end{aligned}$$

Deur  $\sum_{q \in Q_p} [f_q(b_q) - f_q(a_q)] + f_{r_p}(b_{r_p}) - f_{r_p}(x_{r_p})$  weerskante van die ongelykheidsteken af te trek en verder te reduceer, volg dat

$$\begin{aligned} \sum_{j \in J} f_j(b_j) + \sum_{i \in I} f_i(a_i) + f_k(x_k) & > \sum_{j \in J} f_j(b_j) + \sum_{i \in I} f_i(a_i) + f_p(b_p) - f_p(a_p) - \sum_{q \in Q_p} f_q(b_q) \\ & + \sum_{q \in Q_p} f_q(a_q) - f_{r_p}(b_{r_p}) + f_{r_p}(x_{r_p}) + f_k(a_k). \quad (13) \end{aligned}$$

Definieer die nuwe indeksversamelings as  $J^+ = J - Q_p - \{r_p\} + \{p\}$  en  $I^+ = I + Q_p - \{p\} + \{k\}$ . Dan reduceer (13) na

$$\sum_{j \in J} f_j(b_j) + \sum_{i \in I} f_i(a_i) + f_k(x_k) > \sum_{j \in J^+} f_j(b_j) + \sum_{i \in I^+} f_i(a_i) + f_{r_p}(x_{r_p}).$$

'n Nuwe oplossing  $\{I^+, J^+, \{r_p\}\}$  is verkry waarvan die totale opbrengs minder is as dié van die oorspronklike oplossing  $\{I, J, \{k\}\}$ . Die oplossing  $\{I, J, \{k\}\}$  is dus nie optimaal nie en die stelling volg uit die kontrapositiewe. ■

Die stelling dikteer dat elkeen van die nieaktiewe veranderlikes oorweeg moet word om aktief te word en dat dit gedoen moet word as so 'n verandering 'n verbetering in die doelfunksiewaarde teweeg kan bring. Hierdie proses moet voortgesit word totdat daar nie meer 'n enkele nieaktiewe veranderlike is wat 'n verbetering teweeg kan bring nie. Die grootste-verskilalgoritme is hierop gebaseer.

Hierdie stelling reduceer Stelling 3 se boetoplossings (waarop Moré en Vavasis se CKP algoritme gebaseer is) na 'n kleiner versameling boetoplossings deur te vereis dat hulle ook aan voorwaarde (11) voldoen. Dié versameling moet algoritmies verkry word.

### Algoritme 1 (Beginoplossing)

1. **Inisialiseer.**

Stel  $I \leftarrow \{1, \dots, n\}$ ,  $J \leftarrow \emptyset$  en  $H \leftarrow B - \sum_{i \in I} a_i$ . Die indeksversamelings dui die veranderlikes wat by hulle onder- of bogrense is aan:  $x_i = a_i$  vir  $i \in I$ ,  $x_i = b_i$  vir  $i \in J$ . Verder is  $H$  die hoeveelheid van die hulpbron wat nog toegeken moet word.

2. **Ken die hulpbron toe.**

Kies  $j$  sodanig dat

$$\frac{f_j(b_j) - f_j(a_j)}{b_j - a_j} = \min_{i \in I} \frac{f_i(b_i) - f_i(a_i)}{b_i - a_i}$$

en stel  $J \leftarrow J + \{j\}$ ,  $I \leftarrow I - \{j\}$  en  $H \leftarrow H - (b_j - a_j)$ . Herhaal solank die hulpbron genoeg is, dit wil sê totdat 'n  $j$  gevind is waarvoor  $b_j - a_j > H$ .

3. **Kies  $k$  waarvoor  $x_k \in (a_k, b_k)$ .**

Stel  $F = \{i \mid b_i - a_i > H, i \in I\}$  en  $G = \{i \mid b_i - a_i \leq H, i \in I\}$ . Kies  $f$  en  $g$  sodanig dat

$$\frac{f_f(a_f + H) - f_f(a_f)}{H} = \min_{i \in F} \frac{f_i(a_i + H) - f_i(a_i)}{H}$$

en

$$\frac{f_g(b_g) - f_g(a_g)}{b_g - a_g} = \min_{i \in G} \frac{f_i(b_i) - f_i(a_i)}{b_i - a_i}.$$

Indien

$$\frac{f_g(b_g) - f_g(a_g)}{b_g - a_g} \leq \frac{f_f(a_f + H) - f_f(a_f)}{H},$$

stel dan  $I \leftarrow I - \{g\}$ ,  $J \leftarrow J + \{g\}$  en  $H \leftarrow H - (b_g - a_g)$ , andersins stel  $I \leftarrow I - \{f\}$ ,  $k \leftarrow f$  en  $H \leftarrow 0$ .

Herhaal totdat  $H = 0$ . ■

**Algoritme 2 (Grootste-verskilalgoritme (GVA))****1. Toets.**

Deur 'n volledige soekproses met behulp van 'n boomstruktuur, kies vir elke  $p \in I + \{k\}$  'n  $Q_p \subset J$  en  $r_p \in J$ ,  $r_p \notin Q_p$  en sodanig dat (10) geld. Indien die voorwaardes (11) geld, dan is die oplossing bereik — stop die algoritme. Indien (11) nie geld nie, stel

$$V_i \leftarrow \sum_{q \in Q_p} [f_q(b_q) - f_q(a_q)] + f_{r_i}(b_{r_i}) - f_{r_i}(x_{r_i}) \\ - [f_i(b_i) - f_i(a_i)] + [f_k(x_k) - f_k(a_k)]$$

vir alle  $i \in I + \{k\}$ . Kies  $p$  sodanig dat  $V_p = \max_{i \in I \cup \{k\}} V_i$  en gaan na stap (2) vir omruiling.

**2. Omruiling van veranderlikes.**

(a) Indien  $J = \emptyset$  stel dan:  $I \leftarrow I + \{k\} - \{p\}$  en  $k \leftarrow p$ .

(b) Indien  $I = \emptyset$  stel dan:  $J \leftarrow J + \{k\} - \{r\}$  en  $k \leftarrow r$ .

(c) Indien  $I \neq \emptyset$  en  $J \neq \emptyset$  stel dan:  $J \leftarrow J + \{p\} - \{r\} - Q$ ,  $I \leftarrow I + \{k\} - \{p\} + Q$  en  $k = r$ .

Herhaal stappe (1) en (2) totdat die proses konvergeer. ■

**6.1 Konvergensie**

Die algoritme konvergeer in 'n eindige aantal stappe. Met elke nuwe oplossing is daar 'n verbetering van die doelfunksie en geen oplossing kan dus meer as een keer voorkom nie. Die aantal moontlike oplossings is van bo begrens deur  $2^n$ , waar  $n$  die aantal funksies is wat in die doelfunksie gesommeer word, omdat 'n bepaalde indeks 'n element is van óf  $I + \{k\}$  óf  $J$ .

**6.2 Numeriese voorbeeld**

Beskou die vier funksies

$$\begin{aligned} f_1(x_1) &= x_1 - 0,001111x_1^2, & 0 \leq x_1 \leq 20, \\ f_2(x_2) &= 0,6667x_2 - 0,000277x_2^2, & 0 \leq x_2 \leq 80, \\ f_3(x_3) &= 1,5x_3 - 0,0075x_3^2, & 0 \leq x_3 \leq 90, \\ f_4(x_4) &= 1,3x_4 - 0,0065x_4^2, & 0 \leq x_4 \leq 60, \end{aligned}$$

en  $B = 143$ .

**Beginoplossing****1. Inisialiseer**

$I \leftarrow \{1, 2, 3, 4\}$ ,  $J \leftarrow \emptyset$ ,  $H \leftarrow 143$  en  $B \leftarrow 143$ .

2. Ken die hulpbron toe

$$\frac{f_1(20)}{20} = \frac{19,5556}{20} = 0,9778, \quad \frac{f_3(90)}{90} = \frac{74,25}{90} = 0,825,$$

$$\frac{f_2(80)}{80} = \frac{51,5606}{80} = 0,6445, \quad \frac{f_4(60)}{60} = \frac{54,6}{60} = 0,91.$$

Nadat die hulpbron toegeken is soos beskrywe in stap 2 van die beginoplossing, is  $J = \{2, 4\}$ ,  $I = \{1, 3\}$  en  $H = 3$ .

3. Kies  $k$  waarvoor  $x_k \in (a_k, b_k)$ .  $F \leftarrow \{1, 3\}$  en  $G \leftarrow \emptyset$ . Nou is

$$\frac{f_1(3)}{3} = 0,9966, \quad \frac{f_3(3)}{3} = 1,4775,$$

sodat  $I = \{3\}$ ,  $J = \{2, 4\}$  en  $k = 1$ .

Die beginoplossing word deur hierdie  $\{I, J, \{k\}\}$  gegee.

### GVA-oplossing

1. *Toets.* Nou moet  $p$  uit  $I + \{k\} = \{1, 3\}$  gekies word:

2.  $p = 1$ , in hierdie geval het ons twee moontlike  $Q_p$ 's, te wete  $Q_p = \{2\}$  en  $Q_p = \{4\}$ .

- $Q_p = \{2\}$ ,  $r = 4$  en  $j = 4$ :

$$\begin{aligned} x_4 &= (b_q - a_q) + b_j - (b_p - a_p) + (x_k - a_k) \\ &= (b_2 - a_2) + b_4 - (b_1 - a_1) + (x_1 - a_1) \\ &= (80 - 0) + 60 - (20 - 0) + (3 - 0) \\ &= 123 > b_4 \end{aligned}$$

en dus 'n ongeldige opsie.

- $Q_p = \{4\}$ ,  $r = 2$  en  $j = 2$ :

$$\begin{aligned} x_2 &= (b_q - a_q) + b_j - (b_p - a_p) + (x_k - a_k) \\ &= (b_4 - a_4) + b_2 - (b_1 - a_1) + (x_1 - a_1) \\ &= (60 - 0) + 80 - (20 - 0) + (3 - 0) \\ &= 123 > b_2 \end{aligned}$$

en dus 'n ongeldige opsie.

3.  $p = 3$ , in hierdie geval het ons weer twee moontlike  $Q_p$ 's,  $Q_p = \{2\}$  en  $Q_p = \{4\}$ .

- $Q_p = \{2\}$ ,  $r = 4$  en  $j = 4$ :

$$\begin{aligned} x_4 &= (b_q - a_q) + b_j - (b_p - a_p) + (x_k - a_k) \\ &= (b_2 - a_2) + b_4 - (b_3 - a_3) + (x_1 - a_1) \\ &= (80 - 0) + 60 - (90 - 0) + (3 - 0) \\ &= 53 < b_4 \end{aligned}$$

en in hierdie geval is die verbetering

$$f_2(b_2) + f_4(b_4) - f_4(x_4) - f_3(b_3) + f_1(x_1) = -15,7383.$$

- $Q_p = \{4\}$ ,  $r = 2$  en  $j = 2$ :

$$\begin{aligned}
 x_2 &= (b_q - a_q) + b_j - (b_p - a_p) + (x_k - a_k) \\
 &= (b_4 - a_4) + b_2 - (b_3 - a_3) + (x_1 - a_1) \\
 &= (60 - 0) + 80 - (90 - 0) + (3 - 0) \\
 &= 53 < b_2
 \end{aligned}$$

en in hierdie geval is die verbetering

$$f_4(b_4) + f_2(b_2) - f_2(x_2) - f_3(b_3) + f_1(x_1) = 0,346\,194.$$

Die laaste geval gee die grootste verbetering. Dus is  $p = 3$ ,  $Q_p = \{4\}$ ,  $r = 2$ .

#### 4. Omruiwing van veranderlikes

$$\begin{array}{ll}
 J \leftarrow J + \{p\} - \{r\} - Q & I \leftarrow I + \{k\} - \{p\} + Q \\
 = \{2, 4\} + \{3\} - \{2\} - \{4\} & = \{3\} - \{3\} + \{1\} + \{4\} \\
 = \{3\} & = \{1, 4\}
 \end{array}$$

en  $k = r = 2$ .

## 7 Numeriese vergelyking

Om die numeriese resultate te verkry is die verskillende algoritmes in Pascal geprogrammeer en (gedurende 1993) op 'n rekenaar met 'n 386DX verwerker uitgevoer.

### 7.1 Beginoplossings

Moré en Vavasis (1991) volg Stelling 3 direk en gebruik die kleinste hellings by die ondergrense,  $f'_i(a_i)$ ,  $i = 1, 2, \dots, n$ , as kriterium by die keuse van funksies wat by hulle bogrense vasgestel moet word. Die kleinste gemiddelde hellings van die funksies  $[f_i(b_i) - f_i(a_i)] / [b_i - a_i]$ ,  $i = 1, 2, \dots, n$ , is egter 'n beter keuse in die gulsige sin. Dink byvoorbeeld aan twee funksies uit die oorsprong met ondergrense beide nul en dieselfde bogrens. As funksie 1 se helling by sy ondergrens kleiner is as dié van funksie 2, maar sy waarde by die gemeenskaplike bogrens is hoër (die twee sny) dan is die totale opbrengs per hulpbroneenheid by sy bogrens groter is as dié van funksie 2. Funksie 2 is dus 'n beter keuse om by sy bogrens te wees.

In 'n proefneming met 10 funksies en 1 000 verskillende toekennings van hulpbronne (en gevolglik 1 000 verskillende berekenings van die beginoplossing) is daar gevind dat indien die kriterium van kleinste  $f'_i(a_i)$ ,  $i = 1, 2, \dots, n$ , gebruik word, Moré en Vavasis se CKP algoritme die algehele minimum in net 40% van die lopies verkry het. Hierteenoor het die gulsige beginoplossing in 73% van die lopies die algehele minimum gevind. Herhaal op 'n tweede stel funksies was dit 49% teenoor 64%, steeds vór die gulsige beginoplossing. Deur die grootste-verskilalgoritme op die gulsige beginoplossings te gebruik, is die algehele minimum in 100% van die gevalle verkry.

	Berekeningstyd (sekondes per 1 000 toekennings)			
	DP	CKP	GV100	GVA
	<b>Kwadratiese funksies</b>			
Versameling 1	62,60	5,82	1,80	7,39
Versameling 2	62,60	5,76	1,75	7,36
Versameling 3	62,60	5,82	1,60	7,20
Versameling 4	61,68	5,87	1,55	7,40
<b>Eksponensiële funksies</b>				
Versameling 1	59,53	5,87	1,70	7,00
Versameling 2	59,70	5,87	1,40	6,68
Versameling 3	59,53	5,87	1,90	7,30
Versameling 4	59,59	5,87	1,55	7,61
<b>C-op-M funksies</b>				
Versameling 1	59,75	5,87	1,75	7,27
Versameling 2	59,86	5,87	2,00	7,19
Versameling 3	59,64	5,93	2,50	7,60
Versameling 4	59,70	5,87	1,65	7,68
<b>Logaritmiese funksies</b>				
Versameling 1	59,15	5,87	1,65	7,23
Versameling 2	59,59	5,82	1,50	7,67
Versameling 3	59,48	5,82	1,60	6,91
Versameling 4	59,42	5,82	1,40	7,18
Gemiddeld	60,28	5,85	1,71	7,29
DP/Ander algoritme		10,30	35,33	8,27

**Tabel 1:** Berekeningstyd vir verskillende algoritmes.

Aantal Funksies	DP Versameling 4	GVA Versameling 4
0	0,000 00	0,000 00
1	0,001 00	0,000 50
2	0,006 55	0,000 55
3	0,013 33	0,000 93
4	0,019 90	0,001 38
5	0,026 68	0,001 96
6	0,033 50	0,002 48
7	0,040 40	0,003 37
8	0,047 36	0,005 00
9	0,054 56	0,006 19
10	0,061 68	0,007 34

**Tabel 2:** Berekeningstyd met toename in die aantal kwadratiese funksies (sekondes).

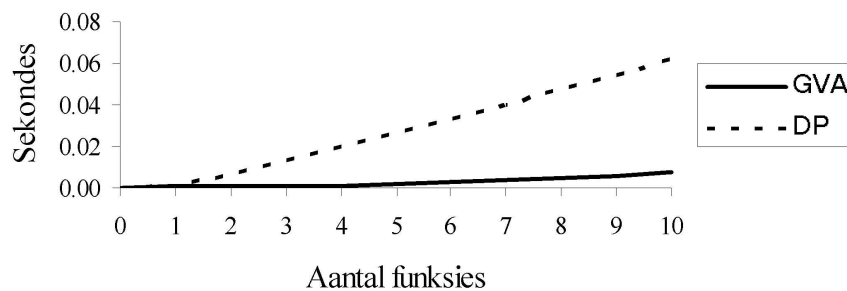
## 7.2 Grootste-verskilalgoritme met vaste intervalle

Vir die probleem (3) is alle  $a_i = 0$  en  $b_i = 100$ ,  $i = 1, 2, \dots, n$ , geneem en verskillende algoritmes is getoets vir vier versamelings van tien elk, vir die vier klasse konkawe funksies (2) soos deur Luss en Gupta (1975) gebruik. Die versamelings is só saamgestel dat sommige versamelings se funksies sny en ander s'n nie.

Die berekeningstyd van die verskillende algoritmes is gemeet oor 1 000 verskillende toekennings van hulpbronne vir 10 funksies. Die resultate word in Tabel 1 weergegee. In die gevalle van die DP-, GV100- en GVA-algoritmes is daar in 100% van die gevalle die algehele minimum verkry, wat nie die geval was met die CKP-algoritme nie. Die kodes DP vir dinamiese programmering (soos in §10.4 van Wagner (1969)), CKP vir die CKP-algoritme van Moré en Vavasis, en GV100 vir die grootste-verskilalgoritme met vaste interval 100, is gebruik. GVA verwys, soos hierbo, na die gewone grootste-verskilalgoritme vir verskillende onder- en bogrense. Die GV100 is gespesialiseer om die enerse definisiegebiede van  $[0, 100]$  wat hier gebruik is, uit te buit. Alhoewel die CKP 'n ietwat vinniger as die GVA is, word sy gebruik uitgesluit deur sy onvermoë (soos hierbo gerapporteer) om selfs in hierdie toetsvoorbeelde die algehele minimum konsekwent op te spoor. Verdere vergelykings is tussen dinamiese programmering (DP) en die GVA.

## 7.3 Toename in aantal funksies

Die DP-algoritme van Wagner (1969) is weereens gebruik om die eksakte oplossings te verkry vir verskillende aantalle funksies (die aantal funksies  $n$  is vermeerder) en die GVA-algoritme se antwoorde is hiermee vergelyk. Die resultate word in Tabel 2 en Figuur 1 weergegee. Die berekeningstyd in die tabel word gegee in sekondes per toekenning (een lopies). Soos gesien kan word, is die GVA aansienlik vinniger as DP. As die eerste twee resultate in die tabel as onbetroubaar weggelaat word, is die gemiddelde berekeningstyd vir die GVA omtrent 'n twaalfde van die DP s'n.



**Figuur 1:** Berekeningstyd met toename in aantal funksies.



### 7.4 Berekeningstyd per enkele toekenning

Vir 'n vaste hulpbrontoekenning van 550 is die DP- en GVA-algoritmes getoets vir 10 funksies met verskillende bogrense. Die berekeningstyd is gemeet in sekondes per enkele toekenning van 550. Soos uit Tabel 3 gesien kan word, is die gemiddelde berekeningstyd van die DP-algoritme 31,828 sekondes terwyl dit vir die GVA-algoritme slegs 0,027 332 sekondes is. Die GVA-algoritme is met 'n enkele hulpbrontoekenning dus gemiddeld 1 164 keer vinniger as die DP-algoritme terwyl dit in 100% van die gevalle die algehele minimum bereken het.

Vir 'n soortgelyke eksperiment met 'n enkele hulpbrontoekenning van 550 vir 10 funksies, maar met verskillende ondergrense, was die GVA-algoritme gemiddeld 268 keer vinniger as die DP-algoritme.

Som van bogrense	DP	GVA	DP / GVA
550	17,52	0,004 50	3 893
550	19,27	0,004 51	4 273
643	24,82	0,024 28	1 022
645	24,99	0,039 55	632
680	28,50	0,088 43	322
700	30,40	0,019 33	1 573
750	34,27	0,029 44	1 164
775	37,62	0,019 72	1 908
813	40,37	0,019 61	2 059
1 000	60,52	0,023 95	2 527
Gemiddeld	31,828	0,027 332	1 164

**Tabel 3:** Berekeningstyd vir enkele toekenning (in sekondes) vir kwadratiese funksies van versameling 3, met  $B = 550$ .

## 8 Maksimering van konvekse funksies

Indien  $f(x)$  'n konvekse funksie is, dan is  $-f(x)$  konkaf (Luenberger (1984)). Om die konvekse funksies  $f(x)$  te maksimeer sal dan dieselfde wees as om die konkawe funksies  $-f(x)$  te minimeer. Dieselfde in die sin dat die optimum in dieselfde punt bereik word. By die konkawe minimeringsprobleem is egter deurlopend aangeneem dat die funksies nedalend is en dus dat hulle negatiewes niestygend is, terwyl die praktiese konvekse maksimeringsprobleem se funksies ook nedalend is. Die proses wat hier gevolg is, moet dus herhaal word en die resultaat is feitlik identies, behalwe dat die algoritmes gewysig word deur maksimums met minimums te vervang en omgekeerd.

## 9 Gevolgtrekking

Om die algemene kontinue knapsakprobleem (1) op te los, word eers na die geval gekyk waar al die funksies konkaf is. Dié probleem is vir alle praktiese doeleindes opgelos.

Daarna word na die geval gekyk waar al die funksies konveks is. Net Moré en Vavasis (1991) het hierop gewerk, maar op die ekwivalente probleem waar die funksies konkav is en die doelfunksie geminimeer moet word. Hulle begin deur net na 'n streng plaaslike minimum te soek en die moeilikheidsgraad van die probleem te verbeter wat hulle ook bereik met hulle CKP algoritme (wat binne polinomie se tydgrense die plaaslike optimum vind).

Ons vind numeries dat die CKP nie vertrou kan word om die algehele minimum te vind as die funksies sny nie. In Stellings 1, 2 en 3 word bekende (of by implikasie bekende) resultate oor die nodige voorwaardes vir 'n minimum formeel saamgevat. In Stelling 4 voeg ons 'n voorwaarde by en alhoewel ons nie bewys het dat dit 'n voldoende voorwaarde vir optimaliteit is nie, gee die grootste-verskilalgoritme (GVA) wat daarop gebaseer is, in 16 000 lopies op voorkomste van Luss en Gupta (1975) se funksies telkens die algehele minimum. Die berekeningstyd vergelyk goed met dié van die CKP, alhoewel dit effens stadiger is.

'n Gulsige beginoplossing word ook voorgestel en numeriese vergelykings plaas dit voor dié van Moré en Vavasis wat Stelling 3 meer direk volg.

## Verwysings

- [1] BITRAN GR & HAX AC, 1979, *On the solution of convex knapsack problems with bounded variables*, pp. 357–367 in PRÉKOPA A (ED.), *Survey of Mathematical Programming*, North-Holland, Amsterdam.
- [2] CHARNES A & COOPER WW, 1958, *The theory of search: Optimum distribution of search effort*, *Management Science*, **5**, pp. 44–49.
- [3] ERENGUC SS, 1988, *Multiproduct dynamic lot-sizing model with coordinated replenishments*, *Naval Research Logistics Quarterly*, **35**, pp. 1–22.
- [4] GUISEWITE GM & PARDALOS PM, 1990, *Minimum concave cost network flow problems: Applications, complexity, and algorithms*, *Annals of Operations Research*, **25**, pp. 75–100.
- [5] GUISEWITE GM & PARDALOS PM, 1991, *Global search algorithms for minimum concave cost network flow problems*, *Journal of Global Optimization*, **1**, pp. 309–330.
- [6] IBARAKI T & KATO N, 1988, *Resource allocation problems*, The MIT Press, Cambridge (MA).
- [7] KONNO H, 1988, *Minimum concave cost production system: A further generalization of multi-echelon model*, *Mathematical Programming*, **41**, pp. 185–193.
- [8] LUENBERGER DG, 1984, *Linear and non-linear programming*, Addison-Wesley Publishing Company, Reading (MA).
- [9] LOOTSMA FA, MENSCH TCA & VOS FA, 1990, *Multi-criteria analysis and budget reallocation in long-term research planning*, *European Journal of Operational Research*, **47**, pp. 293–305.

- [10] LUSS H & GUPTA SK, 1975, *Allocation of effort resources among competing activities*, Operations Research, **23**, pp. 360–366.
- [11] MCCORMICK GP, 1973, *Algorithmic and computational aspects of the use of optimization methods in engineering design*, Computers & Structures, **3(5)**, pp. 1241–1249.
- [12] MORÉ JJ & VAVASIS SS, 1991, *On the solution of concave knapsack problems*, Mathematical Programming, **49**, pp. 397–411.
- [13] VAN DEN BOSCH PPJ & LOOTSMA FA, 1987, *Scheduling of power generation via large-scale nonlinear optimization*, Journal on Optimization Theory and Applications, **55**, pp. 313–326.
- [14] VEROY B & ZWASS B, 1987, *Embedding a point-to-point network in the expansion of infrastructure for information systems*, Journal of Management Information Systems, **4**, pp. 50–63.
- [15] WAGNER HM, 1969, *Principles of operations research with applications to managerial decisions*, Prentice Hall, Englewood Cliffs (NJ).
- [16] Wilkinson C & Gupta SK, 1969, *Allocating promotional effort to competing activities: A dynamic programming approach*, Proceedings of the 5<sup>th</sup> Triennial Conference of the International Federation of Operations Research Societies, Venice, pp. 419–432.
- [17] ZWASS V & VEROY B, 1988, *Capacity expansion for information flow distribution in multi-path computer communication networks*, Journal of Management Information Systems, **5**, pp. 57–70.

