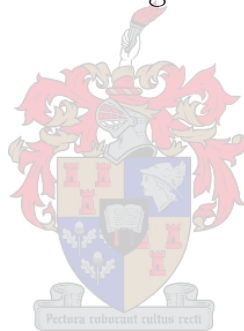


Algoritmes vir die maksimering van konvekse en verwante knapsakprobleme

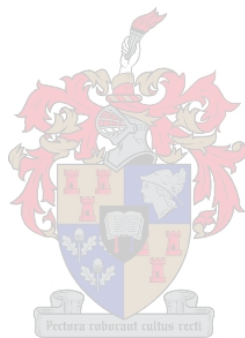
SE Visagie



Proefskrif ingelewer vir die graad Doktor in die Wysbegeerte aan die Universiteit van Stellenbosch.

November 2006

Promotor: Prof. HC de Kock



Verklaring

Ek, die ondergetekende, verklaar hiermee dat die werk in hierdie proefskrif vervat, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik by enige universiteit ter verkryging van 'n graad voorgelê het nie.

S.E. Visagie

Datum





OPSOMMING

In hierdie proefskrif word oorspronklike algoritmes voorgestel vir die oplossing van skeibare hulpbrontoekeningsprobleme (HTP's) met stygende nielineêre doelfunksies, en met onder- en bogrense op elke veranderlike. Algoritmes word voorgestel vir drie spesiale gevalle van HTP's. Die eerste spesiale geval is waar die HTP se doelfunksie uit die som van konvekse funksies bestaan en al die veranderlikes van hierdie funksies oor dieselfde intervallengte strek. In die tweede spesiale geval word ook na HTP's gekyk met die som van konvekse funksies in die doelfunksie, maar al die veranderlikes van hierdie funksies strek oor verskillende intervale. In die laaste spesiale geval word HTP's met 'n doelfunksie wat uit die som van konvekse en konkawe funksies bestaan, beskou. In hierdie geval kan die veranderlikes van hierdie funksies ook oor wisselende intervale strek.

In die eerste geval word twee nuwe algoritmes, naamlik die breuk- en die hellingalgoritme voorgestel vir die oplossing van 'n HTP wat aan hierdie geval se voorwaardes voldoen. Beide hierdie algoritmes lewer ordes beter oplossingstye as die bestaande vertak-en-begrens-algoritme.

'n Nuwe heuristiek en drie nuwe algoritmes word aangebied om die HTP's vir die tweede geval op te los. Die isogrensheuristiek lewer gemiddeld goeie oplossings relatief tot die optimale doelfunksie in baie vinniger oplossingstye as die eksakte algoritmes. Die drie algoritmes, naamlik die isogrens-vertak-en-begrens-algoritme, die vertak-sny-en-begrens-algoritme en die isogrens-vertak-sny-en-begrens-algoritme lewer ook beduidend beter oplossing as die bestaande vertak-en-begrens-algoritme. Dit word aangetoon dat die isogrens-vertak-sny-en-begrens-algoritme gemiddeld die vinnigste oplossingstye het, gevolg deur die isogrens-vertak-en-begrens-algoritme en dan die vertak-sny-en-begrens-algoritme.

In die derde geval word die nodige en voldoende voorwaardes vir optimaliteit in oënskou geneem. Hieruit word die gevolgtrekking gemaak dat daar te veel berekenings nodig is om al die punte wat aan die nodige voorwaardes voldoen, analities op te spoor. Daarom word drie nuwe algoritmes, naamlik die KL, SKL- en die IKL-algoritme aangebied vir die oplossing van HTP's vir hierdie geval. Hierdie algoritmes is veralgemenings van die vertak-en-begrens-algoritme, vertak-sny-en-begrens-algoritme en die isogrens-vertak-en-begrens-algoritme onderskeidelik. Die KL-algoritme is as bakenalgoritme gebruik. Slegs die IKL-algoritme het beduidend korter oplossingstye as die KL-algoritme.

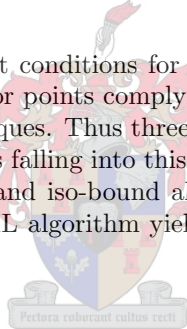
ABSTRACT

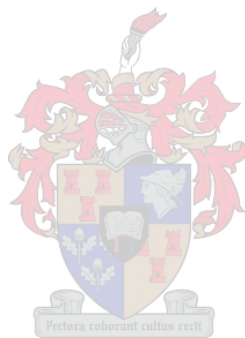
In this dissertation original algorithms are introduced to solve separable resource allocation problems (RAPs) with increasing nonlinear functions in the objective function, and lower and upper bounds on each variable. Algorithms are introduced in three special cases. The first case arises when the objective function of the RAP consists of the sum of convex functions and all the variables for these functions range over the same interval. In the second case RAPs with the sum of convex functions in the objective function are considered, but the variables of these functions can range over different intervals. In the last special case RAPs with an objective function comprising the sum of convex and concave functions are considered. In this case the intervals of the variables can range over different values.

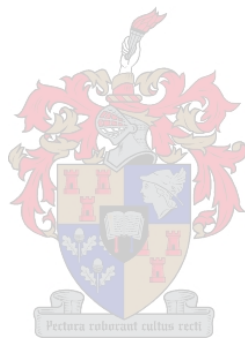
In the first case two new algorithms, namely the fraction and the slope algorithm are presented to solve the RAPs adhering to the conditions of the case. Both these algorithms yield far better solution times than the existing branch and bound algorithm.

A new heuristic and three new algorithms are presented to solve RAPs falling into the second case. The iso-bound heuristic yields, on average, good solutions relative to the optimal objective function value in faster times than exact algorithms. The three algorithms, namely the iso-bound algorithm, the branch and cut algorithm and the iso-bound branch and cut algorithm also yield considerably better solution times than the existing branch and bound algorithm. It is shown that, on average, the iso-bound branch and cut algorithm yields the fastest solution times, followed by the iso-bound algorithm and then by the branch and cut algorithm.

In the third case the necessary and sufficient conditions for optimality are considered. From this, the conclusion is drawn that search techniques for points complying with the necessary conditions will take too long relative to branch and bound techniques. Thus three new algorithms, namely the KL, SKL and IKL algorithms are introduced to solve RAPs falling into this case. These algorithms are generalisations of the branch and bound, branch and cut, and iso-bound algorithms respectively. The KL algorithm was then used as a benchmark. Only the IKL algorithm yields a considerable improvement on the KL algorithm.

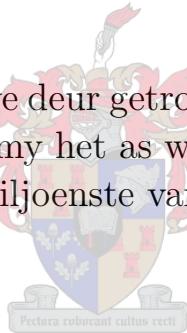




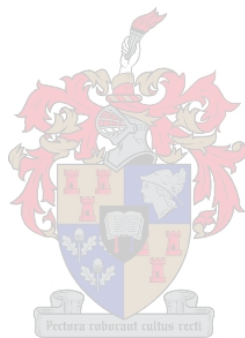


'n Paar groot dankies aan:

- Kobus Wolvaardt,
vir die idee van die onderwerp wat in hierdie proefskrif aangepak is;
- Hennie de Kock,
vir jou hulp, insette, goeie raad en insig oor die jare;
- Heloïse en Willem,
vir julle ondersteuning en omdat julle verstaan dat ek julle saans moes afskeep terwyl ek gewerk het om dié proefskrif klaar te kry;
- pa en ma,
omdat julle nog my hele lewe deur getrou agter my staan en deurentyd meer vertrou en geloof in my het as wat ek ooit in myself sal hê. Ek wens soms ek was net 'n miljoenste van die mens wat julle glo ek is.



Ek dra hierdie proefskrif op aan my gesin en ouers.



Inhoudsopgawe

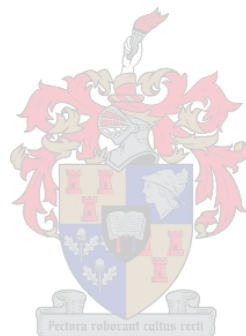
Lys van figure	viii
Lys van tabelle	ix
Notasie	xi
1 Inleiding	1
1.1 Notasie	4
1.2 Toepassings	5
1.2.1 Bemarking	5
1.2.2 Chemie	6
1.2.3 Kapasiteitsbepaling in vervaardiging, gesondheidsorg en rekenaarnetwerke	6
1.2.4 Portefeulje	7
1.2.5 Produksiebeplanning en voorraadbeheer	8
1.2.6 Soektogte	9
1.2.7 Steekproewe	10
1.2.8 Hulpbronverdeling	11
1.3 Doelwitte van die proefskrif	13
1.4 Proefskrifuitleg	13
2 Literatuur	17
2.1 Maksimering van konvekse (minimering van konkawe) doelfunksies	17
2.1.1 Luss en Gupta	18
2.1.2 Moré en Vavasis	18
2.1.3 Venter en Wolvaardt	19

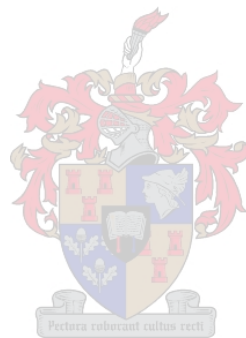
2.1.4	Dai en Fletcher	19
2.1.5	Sun, Wang en Li	19
2.2	Maksimering van konkawe (minimering van konvekse) doelfunksies	20
2.3	Gemengde doelfunksies	20
2.4	Tipes funksies in die doelfunksie	20
2.4.1	Stygende konvekse funksies	21
2.4.2	Stygende konkawe funksies	22
2.5	Dinamiese programmering as oplossingsmetode	23
2.6	Vertak-en-begrens-metodes	24
2.6.1	Gewone vertak-en-begrens-algoritme	24
2.6.2	Lineêre omsluiting	25
2.6.2.1	Omsluit slegs die funksie in versameling \mathcal{K}	26
2.6.2.2	Algemene lineêre omsluiting	27
2.7	Nielineêre optimering	29
2.7.1	Globale optimering	29
2.8	Kuhn-Tucker-voorwaardes vir HTP	32
2.9	Kromming van 'n funksie	34
2.10	Primale simpleksalgoritme met bogrense	35
2.11	Duale simpleksalgoritme met bogrense	37
2.12	Samevatting	37
3	<i>K/E/V</i>-HTP's	39
3.1	Kuhn-Tucker-voorwaardes vir <i>K/E/G</i>	39
3.2	Vaste interval	39
3.3	Heeltallige programmering as oplossingsmetode vir <i>K/E/V</i>	42
3.4	Breukalgoritme as oplossingsmetode vir <i>K/E/V</i>	43
3.5	Hellingalgoritme as oplossingsmetode vir <i>K/E/V</i>	47
3.6	Samevatting	51
3.6.1	Metodologie	51
3.6.2	Resultate	52
3.6.3	Gevolgtrekkings	53

4	<i>K/E/G</i>: Soekalgoritmes	57
4.1	Teoretiese beskouing	57
4.1.1	Tweede orde nodige voorwaardes	57
4.1.2	'n Ander blik op Stelling 3.1	59
4.2	Soekalgoritmes	60
4.2.1	Breukalgoritme [brute krag]	60
4.2.2	Blaarsoekalgoritme	61
4.2.3	Taksoekalgoritme	65
4.3	Samevatting	70
4.3.1	Metodologie	70
4.3.2	Resultate	70
4.3.3	Gevolgtrekkings	71
5	<i>K/E/G</i>: Vertak-en-begrens-algoritmes	73
5.1	Inleiding tot vertak-en-begrens-algoritmes	73
5.2	Definisies	76
5.3	Isogrens-vertak-en-begrens-benadering	78
5.3.1	Isogrensheuristiek	79
5.3.1.1	Iso-ondergrensprosedure	79
5.3.1.2	Iso-bogrensprosedure	82
5.3.1.3	<i>K/E/V</i> -HTP's en die isogrensheuristiek	84
5.3.2	Isogrens-vertak-en-begrens-algoritme	85
5.4	Vertak-sny-en-begrens-algoritme	92
5.5	Isogrens-vertak-sny-en-begrens-algoritme	103
5.5.1	Numeriese metode om w_r te bepaal	107
5.6	Rangorde en aantal eksperimente	108
5.7	Samevatting	110
5.7.1	Metodologie	110
5.7.2	Resultate	111
5.7.2.1	Invloed van gemiddelde helling	113
5.7.2.2	Prestasie met die oplossing van <i>K/E/V</i> -HTP	115
5.7.2.3	Prestasie van vertak-en-begrens-algoritmes vs isogrensheuristiek vir <i>K/E/G</i> -HTP's	117

5.7.2.4	Prestasie van isogrensheuristiek vir $K/E/V$ -HTP's . . .	118
5.7.3	Gevolgtrekkings	119
6	$K/E, A/G$: Kontinue benadering	123
6.1	Teoretiese beskouing van een konkawe en een konvekse funksie	123
6.1.1	$x + y = B$ die enigste bindende beperking	124
6.1.2	$x + y = B$ en $0 \leq x \leq b_x$ is bindend	125
6.1.2.1	$x + y = B$ is bindend en $x = b_x$	125
6.1.2.2	$x + y = B$ is bindend en $x = 0$	127
6.1.3	$x + y = B$ en $0 \leq y \leq b_y$ is bindend	128
6.1.3.1	$x + y = B$ is bindend en $y = b_y$	128
6.1.3.2	$x + y = B$ is bindend en $y = 0$	129
6.1.4	Optimaliteit vir twee funksies	129
6.1.5	Herlei na 'n funksie in een veranderlike	131
6.2	Veralgemening na meer funksies	132
6.3	Gevolgtrekkings	136
7	$K/E, A/G$: Vertak-en-begrens-algoritmes	137
7.1	KL-algoritme (vertak-en-begrens-algoritme)	137
7.2	SKL-algoritme (vertak-sny-en-begrens)	139
7.3	IKL-algoritme (isogrens-vertak-en-begrens)	143
7.4	Samevatting	147
7.4.1	Metodologie	148
7.4.2	Resultate	148
7.4.2.1	Verskillende verhoudings van konkawe en konvekse funksies	148
7.4.2.2	Invloed van gemiddelde helling	150
7.4.2.3	Invloed van konvekse tob-funksie	152
7.4.3	Gevolgtrekkings	153
8	Slotopmerkings	155
8.1	Proefskrifopsomming	155
8.2	Samevatting en gevolgtrekking	156
8.3	Validering van algoritmes	157

8.4	Idees vir verdere studie	158
8.4.1	Idees vir die uitbreiding van hierdie studie	158
8.4.2	Idees vir verdieping van sekere aspekte in hierdie studie	159
8.5	Bereiking van doelwitte	160
Verwysings		163
A Kuhn-Tucker-voorwaardes		167





Lys van Figure

2.1	Die vertakking van funksie k in twee subprobleme.	26
2.2	Voorbeelde van tipes ekstreme punte.	30
3.1	Die liniëring van $f_i(x_i)$	41
3.2	Gemiddelde oplossingstye vir 'n $K/E/V$ -HTP teenoor n	53
3.3	Die oplossingstye van BrkAlg en HelAlg relatief tot die vertak-en-begrens-algoritme.	55
3.4	Oplossingstye vir 'n $K/E/V$ -HTP met dinamiese programmering teenoor B	55
4.1	Die totale oplossingstye van 100 $K/E/G$ -HTP's teenoor n	71
5.1	Grafiese voorstelling van die werking van die iso-ondergrensprosedure. . .	82
5.2	Grafiese voorstelling van die werking van die iso-ondergrensprosedure (vervolg. . .).	83
5.3	Vertak-en-begrens-boom van die isogrens-vertak-en-begrens-algoritme. . .	92
5.4	Vertak-en-begrens-boom van die isogrens-vertak-en-begrens-algoritme nadat iso-bogrensalgoritme uitgevoer is.	93
5.5	Die bepaling van 'n ondersnit.	94
5.6	Skematiese voorstelling van die geval waar $v_r \in (l_k, R)$ en $w_r \in (R, u_k)$. .	98
5.7	Skematiese voorstelling van die geval waar $v_r = l_r$ en $w_r \in [R, u_k)$	99
5.8	Skematiese voorstelling van die geval waar $v_r \in (l_k, R]$ en $w_r = u_k$	99
5.9	Skematiese voorstelling van die geval waar $v_r = l_k$ en $v_r = u_k$	100
5.10	Die totale oplossingstyd van 100 $K/E/G$ -HTP's teenoor n	112
5.11	Relatiewe oplossingstye vir $K/E/G$ -HTP's teenoor n	112
5.12	Aantal subprobleem vir $K/E/G$ -HTP's teenoor n	113
5.13	Relatiewe hoeveelheid subprobleme om $K/E/G$ -HTP's op te los teenoor n . .	113
5.14	Relatiewe oplossingstye per subprobleem vir $K/E/G$ -HTP's teenoor n . .	114

5.15	Relatiewe oplossingstye vir $K/E/G$ -HTP's teenoor Ω	115
5.16	Totale oplossingstye vir $K/E/V$ -HTP's met $K/E/G$ -algoritmes teenoor n	116
5.17	Gemiddelde persentasie fout in doelfunksiewaarde van die isogrensheuristiek vir $K/E/G$ -HTP's teenoor n	118
5.18	Oplossingstye van die isogrensheuristiek vir 100 lukraak gegenerende $K/E/G$ -HTP's.	120
5.19	Oplossingstye van IGH, BrkAlg en HelAlg vir 100 lukraak gegenerende $K/E/V$ -HTP's teenoor n	121
7.1	Skematiese voorstelling van die geval waar $R < f_r^{-1}(g_{h_r})$	140
7.2	Skematiese voorstelling van die geval waar $R > f_r^{-1}(g_{h_r})$	141
7.3	Die totale oplossingstye om 100 lukraak gegenerende $K/E, A/G$ -HTP's wat elkeen 60 funksies in die doelfunksie, waarvan A_{kv} konvekse funksies is.	149
7.4	Die totale aantal subprobleme om 100 lukraak gegenerende $K/E, A/G$ -HTP's wat elkeen 60 funksies in die doelfunksie, waarvan A_{kv} konvekse funksies is.	150
7.5	Die relatiewe oplossingstye om 100 lukraak gegenerende $K/E, A/G$ -HTP's wat elkeen 60 funksies in die doelfunksie, waarvan A_{kv} konvekse funksies is.	151
7.6	Die relatiewe aantal subprobleme om 100 lukraak gegenerende $K/E, A/G$ -HTP's wat elkeen 60 funksies in die doelfunksie, waarvan A_{kv} konvekse funksies is.	151
7.7	Die totale oplossingstye om 100 lukraak gegenerende $K/E, A/G$ -HTP's teenoor intervallengte Ω	152

Lys van Tabelle

1-1	Verskillende waardes wat $X/Y/Z$ kan aanneem.	4
3-1	Totale oplossingstyd in sekondes van 100 lukraak gegenerende $K/E/V$ -HTP's.	52
3-2	Oplossingstye vir die breuk- en hellingalgoritme.	54
3-3	Invloed van vaste intervallenlengte op oplossingstye van die breuk- en hellingalgoritme.	54
4-1	Die oplossingstye in sekondes van 100 lukraak gegenerende $K/E/G$ -HTP's.	70
5-1	Tabel vir die bepaling van Dudewicz en Dalal se konstante.	109
5-2	Rangorde van oplossingstye vir VSB, IGVSb, VB en IGVB.	110
5-3	Die totale oplossingstyd in sekondes van $K/E/G$ -HTP's teenoor n	111
5-4	Die totale oplossingstyd om 100 lukraak gegenerende $K/E/G$ -HTP's teenoor Ω	114
5-5	Die totale oplossingstyd in sekondes van 100 lukraak gegenerende $K/E/V$ -HTP's teenoor n	116
5-6	Gemiddelde persentasie fout van die isogrensheuristiek.	117
5-7	Die oplossingstye van die isogrensheuristiek vir 100 lukraak gegenerende $K/E/G$ -HTP's teenoor n	119
5-8	Die oplossingstye van die IGH, BrkAlg en HelAlg vir 100 lukraak gegenerende $K/E/V$ -HTP's teenoor n	121
7-1	Die totale oplossingstyd in sekondes van 100 lukraak gegenerende $K/E, A/G$ -HTP's teenoor A_{kv}	147
7-2	Die totale aantal subprobleme om 100 lukraak gegenerende $K/E, A/G$ -HTP's op te los teenoor A_{kv}	149
7-3	Die totale oplossingstyd vir 100 lukraak gegenerende $K/E, A/G$ -HTP's teenoor Ω	152



Notasie

Akronieme

Akroniem	Beskrywing
BGK	Bogrenskind
BGKP	Bogrenskindprobleem
BSKP	Bo-snitkindprobleem
BSP	Bo-snitprobleem
HTP	Hulpbrontoekenningsprobleem
LP	Lineêre programmering
NLPP	Nielineêre programmeringsprobleem
OGK	Ondergrenskind
OGKP	Ondergrenskindprobleem
OSKP	Ondersnitkindprobleem
OSP	Ondersnitprobleem
SP	Snitprobleem
tob	Tussen onder- en bogrens
VBA	<i>Visual Basic for Applications</i>

Gereserveerde simbole

Simbool	Beskrywing
\mathcal{A}	Versameling van al die gesorteerde funksies
A_b	Aantal funksies ($f_i(x_i)$) wat 'n toedeling gelyk aan hul bogrens kry
A_{kv}	Aantal konvekse funksies
A_{kk}	Aantal konkawe funksies
b_1	n -dimensionele binêre vektor met komponente $(b_0, b_1, \dots, b_{n-1})$
B_t	Hoeveelheid hulpbron beskikbaar vir toedeling
B	Hoeveelheid hulpbron beskikbaar
$B_h(i)$	Binêre veranderlikes wat gebruik word om kandidate vir die tob-funksie te identifiseer
B^k	Regterkant van die hulpbronbeperking van die k -de subprobleem
B_t	Tydlike hoeveelheid hulpbron

Simbool	Beskrywing
B_{kk}	Gedeelte van die hulpbron wat aan die konkawe funksies toegeedeel word
B_{kv}	Gedeelte van die hulpbron wat aan die konvekse funksies toegeedeel word
\mathcal{D}	Die versameling van subprobleme wat met die IBKL-prosedure opgelos moet word
D	Indeks oor die versameling \mathcal{D}
D_{ap}	Aantal elemente in versameling \mathcal{D}
D_b	Beste (baken-) doelfunksiewaarde van al die subprobleme wat reeds opgelos is (Eng.: <i>benchmark</i>)
D_i	Optimale doelfunksiewaarde met funksie i as die tob-funksie
D_m	Die beste doelfunksiewaarde tot op hede
D_t	Tydlike doelfunksiewaarde tydens die toedeling
$D_w(\cdot)$	Werklike doelfunksiewaarde van 'n probleem met oplossing \cdot
$D_L(\cdot)$	Lyndoelfunksiewaarde van 'n probleem met oplossing \cdot
\underline{e}_i	n -dimensionele binêre eenheidsvektor. Die enkele komponent wat waarde 1 het, is die i -de komponent. Dit wil sê $\underline{e}_i = (e_0, e_1, \dots, e_{i-1}, e_i, e_{i+1}, \dots, e_{n-1}) = (0, 0, \dots, 0, 1, 0, \dots, 0)$.
$f_i(\cdot)$	Die i -de funksie in die doelfunksie van 'n HTP
\mathcal{G}_b	Bogrensversameling
\mathcal{G}_o	Ondergrensversameling
GRT	Maksimum beginhelling van alle konkawe funksies, <i>i.e.</i> $\max_i[f_i(l_i)]$
H	Konsensushelling
i, j, k, ℓ	Indekse (tellers)
$I_b(i)$	Wyser vir gesorteerde funksies volgens bohelling
$I_g(i)$	Wyser vir gesorteerde funksies volgens gemiddelde helling
$I_o(i)$	Wyser vir gesorteerde funksies volgens onderhelling
\mathcal{I}^o	Iso-onderversameling
\mathcal{I}^b	Iso-boversameling
$I_t(i)$	Wyser vir nie-tob-funksies
$Kand(i)$	Wyser vir kandidate vir die tob-funksie
KLN	Minimum eindhelling van alle konkawe funksies, <i>i.e.</i> $\min_j[g_j(u_j)]$
l_i	Ondergrens van die x_i
l_i^k	Ondergrens van die i -de veranderlike in die k -de subprobleem
\mathcal{N}^{b, H_t}	Bogrensnageslag
\mathcal{N}^{o, H_t}	Ondergrensnageslag
$O_h(i)$	Binêre veranderlike wat gebruik word om kandidate vir die tob-funksie te identifiseer
\mathcal{P}	Die versameling van subprobleme wat met die IOKL-prosedure opgelos moet word
P	Indeks oor die versameling \mathcal{P}
P_{ap}	Aantal elemente in versameling \mathcal{P}

Notasie	Beskrywing
$R_s(i)$	Die hoeveelheid hulpbron wat oor is om toe te deel nadat funksie i sy toedeling gekry het
u_i	Bogrens van x_i
u_i^k	Bogrens van die i -de veranderlike in die k -de subprobleem
\underline{x}	Oplossingsvektor met komponente (x_1, x_2, \dots, x_n)
\underline{x}^b	Beste oplossing
x_i	i -de veranderlike van 'n HTP
\underline{x}^m	Die beste (maksimum doelfunksiewaarde) oplossing $\underline{x}^m = (x_1^m, x_2^m, \dots, x_i^m, \dots, x_n^m)$
\underline{x}^t	Tydlike oplossing, met elemente $(x_1^t, x_2^t, \dots, x_n^t)$
z_i	i -de inskrywing in die z -ry van die simplekstablo





Hoofstuk 1

Inleiding

Die knapsakprobleem is 'n baie bekende probleem in operasionele navorsing. Dit kan beskryf word as 'n wiskundige programmeringsprobleem met een doelfunksie en een beperking. Indien beide die doelfunksie en die beperking lineêr is, staan dit as 'n lineêre knapsakprobleem bekend. Indien een of beide van hierdie funksies nielineêr is, gee dit aanleiding tot 'n nielineêre knapsakprobleem. Daar is baie tipes en toepassings van knapsakprobleme (sien byvoorbeeld [16]). Die knapsakprobleem kan beskou word as 'n probleem waar 'n enkele hulpbron (een beperking) optimaal, dit wil sê volgens die enkele doelfunksie, toegedeel moet word. In die klassieke knapsakprobleem is die skaars hulpbron natuurlik die ruimte in die knapsak en die toedeling moet optimaal geskied volgens die nut of nuttigheid van die items wat hierdie ruimte opgebruik. Op hierdie manier beskou, kan die knapsakprobleem ook 'n hulpbrontoekenningsprobleem (HTP) genoem word. Dit blyk uit meer onlangse literatuur dat 'n HTP in die algemeen gedefinieer word as 'n skeibare knapsakprobleem [5, 6, 31]. In hierdie proefskrif word slegs skeibare knapsakprobleme beskou en daarom word die benaming van HTP in hierdie proefskrif gebruik.

'n Knapsakprobleem kan wiskundig beskryf word as die probleem om 'n hulpbron, waarvan daar in totaal B is, te verdeel sodat die doelfunksie $f(\underline{x})$, met $\underline{x} = (x_1, x_2, x_3, \dots, x_n)$, so 'n groot of so 'n klein as moontlike waarde sal aanneem. Aanvaar die hulpbron moet opgebruik word. Die wiskundige formulering volg dan as

$$\text{min/maks } f(\underline{x})$$

onderhewig aan

$$\begin{aligned}g(\underline{x}) &= B \\ \underline{x} &\geq \underline{0}.\end{aligned}$$

In baie gevalle is $g(\underline{x})$ 'n skeibare lineêre funksie waarin elke x_i 'n koëffisiënt van 1 het, sodat die enkele beperking ook met

$$\sum_{i=1}^n x_i = B$$

vervang kan word. In baie van hierdie gevalle kan die doelfunksie ook geskei word. Dus volg dit dat

$$f(\underline{x}) = \sum_{i=1}^n f_i(x_i).$$

Die doelfunksie kan geïnterpreteer word as 'n koste, verlies, wins, inkomste of uitbetaling wat geskied volgens die toekenning van die hulpbron (byvoorbeeld begrotings, grondstowwe, mense of masjiene). In die geval van koste of verlies is die logiese benadering om die doelfunksie te minimeer terwyl maksimering die logiese keuse is in die geval van wins of inkomste. Die verskil tussen maksimering en minimering is egter nie groot nie, aangesien die maksimering van f ekwivalent is aan die minimering van $-f$. Die veranderlike x_i verteenwoordig die hoeveelheid van die hulpbron wat aan aktiwiteit i toegeken word. Indien die hulpbron deelbaar is, byvoorbeeld wanneer die hulpbron geld is, is dit 'n kontinue knapsakprobleem en kan x_i enige positiewe waarde aanneem. Indien die hulpbron nie deelbaar is nie, byvoorbeeld wanneer dit mense, masjiene, ens. voorstel, sal x_i 'n diskrete veranderlike wees en kan dit slegs positiewe heelgetalle aanneem. Die oplossingsmetodes vir die kontinue en diskrete probleem verskil noemenswaardig. In hierdie proefskrif word aandag geskenk aan die kontinue probleem.

In die algemeen kan daar onder- en bogrense geplaas word op die waardes wat veranderlikes x_i kan aanneem. Dit sou beteken dat dit soms nodig mag wees dat daar ten minste l_i , maar nie meer as u_i nie, van die hulpbron aan aktiwiteit i toegeken kan word. Hierdie grense kan in die wiskundige formulering bygevoeg word as

$$l_i \leq x_i \leq u_i \quad i = 1, 2, 3, \dots, n.$$

Daar is min knapsakprobleme waarin die algemene vorm, $f(x_1, x_2, x_3, \dots, x_n)$, van die doelfunksie voorkom. Die doelfunksie het gewoonlik een of ander spesiale vorm wat gebruik kan word om die probleem op te los. Die tipiese vorms wat in die geval van kontinue probleme voorkom, is:

1. *Skeibaar*: In hierdie geval kan die doelfunksie geskei word in n verskillende funksies wat elk 'n funksie van slegs een veranderlike is. Dit beteken

$$f(x_1, x_2, x_3, \dots, x_n) = \sum_{i=1}^n f_i(x_i).$$

Die volgende drie gevalle kan hier ontstaan [7]:

- (a) *Skeibaar en konveks*: Dit beteken dat elkeen van die funksies $f_i(x_i)$ wat in die doelfunksie is, 'n konvekse funksie is.
- (b) *Skeibaar en konkaf*: Dit beteken dat elkeen van die funksies $f_i(x_i)$ wat in die doelfunksie is, 'n konkawe funksie is.
- (c) *Skeibaar en gemeng*: Dit beteken dat die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$ wat in die doelfunksie is, konkaf of konveks of S-vorming kan wees.

2. *Nieskeibaar*: In hierdie geval kan die doelfunksie nie geskei word in funksies van 'n enkele veranderlike nie. Hier kan ook drie gevalle geïdentifiseer word:

- (a) *Nieskeibaar en konveks*: Dit beteken dat die doelfunksie konveks is.
- (b) *Nieskeibaar en konkav*: Dit beteken dat die doelfunksie konkav is.
- (c) *Nieskeibaar en algemeen*: Dit beteken dat die doelfunksie nie konveks of konkav is nie. Dit kan wel gebeur dat dit in sekere gebiede konkav of konveks kan wees.

Die probleme wat in hierdie proefskrif aangepak word, is dan spesifiek die probleme wat in punt 1 hierbo gegee word en daar word voortaan na hierdie probleem as 'n HTP verwys. Die doelfunksie kan geskei word in óf 'n versameling stygende konvekse funksies óf 'n versameling stygende funksies wat óf konveks óf konkav is. Elkeen van die veranderlikes het onder- en bopgrense op die waardes wat dit mag aanneem. Die bo- en ondergrense kan ook beskou word as ekstra beperkings. Daar word ook aanvaar dat die eerste en tweede afgeleides van al die funksies in die doelfunksie bestaan en kontinu is oor die interval $[l_i, u_i]$, dit wil sê $f_i(x_i) \in C^2$ vir $i = 1, 2, \dots, n$. Die eerste afgeleide word net in die gemengde geval (Hoofstukke 6 en 7) gebruik.

Die algemene formulering vir die probleem wat in hierdie proefskrif aangespreek word, is dus

$$\text{maks} \quad \sum_{i=1}^n f_i(x_i)$$



onderhewig aan

$$\begin{aligned} \sum_{i=1}^n x_i &= B \\ -x_i &\leq -l_i & i = 1, 2, \dots, n \\ x_i &\leq u_i & i = 1, 2, \dots, n \end{aligned} \quad (1.1)$$

Verder word vereis dat

$$\sum_{i=1}^n l_i < B < \sum_{i=1}^n u_i$$

en

$$l_i \geq 0 \quad i = 1, 2, \dots, n.$$

Indien B kleiner as die som van die ondergrense of groter as die som van die bopgrense is, is geen toelaatbare oplossing moontlik nie. Indien B gelyk is aan die som van die ondergrense, is daar net een triviale oplossing moontlik, naamlik $\underline{x} = (l_1, \dots, l_n)$. Net so word die enigste oplossing indien B gelyk is aan die som van die bopgrense, gegee deur $\underline{x} = (u_1, \dots, u_n)$. In beide hierdie gevalle impliseer dit dat geen optimalisering moontlik is nie omdat daar geen alternatiewe beskikbaar is nie. In die hoofstukke wat volg, word verskillende vorms van die funksies $f_i(x_i)$ beskou.

Ibaraki en Katoh [16] toon aan dat indien 'n $l_i > 0$ kan die probleem na 'n ekwivalente probleem getransformeer word sodat sodat alle $l_i = 0$. Daar kan ook 'n transformasie uitgevoer word wat sal verseker dat $f_i(0) = 0$. Dit wil sê al die funksies in die doelfunksie van die HTP kan getransformeer word sodat hulle in die oorsprong begin. In die res van hierdie proefskrif word in die ontwikkeling van die algoritmes aanvaar dat al die funksies $f_i(x_i)$ die eienskap het dat $f_i(0) = 0$ en $l_i = 0$. Anders gestel beteken dit dat in al die algoritmes aanvaar word dat die nodige transformasies (sodat al die funksies $f_i(x_i)$ in die oorsprong begin) reeds gedoen is voordat die algoritme begin.

1.1 Notasie

Daar word na spesiale gevalle van die algemene formulering in (1.1) gekyk. Om vinnig te kan verwys na watter spesiale geval ter sprake is, word die volgende notasie gebruik:

$$X/Y/Z$$

Die eerste eienskap van die probleem waarvoor uitspraak gelewer moet word, is of die doelfunksie gemaksimeer of geminimeer moet word. Dit word in posisie X aangetoon. Die tweede eienskap waarvoor uitspraak gelewer moet word, is die aard van die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$. Die som van hierdie funksies is die doelfunksie. Dit word in posisie Y aangetoon. Die derde eienskap is die intervalle waartoe die veranderlikes x_i met $i = 1, 2, \dots, n$ beperk word. Dit word deur posisie Z aangetoon.

Die waardes wat X , Y en Z kan aanneem, word in Tabel 1-1 gegee.

X	Y	Z
K - maksimeer	A - konkaf	V - vaste interval
N - minimeer	E - konveks	G - enige interval
	S - S-kurwe	

Tabel 1-1: Verskillende waardes wat $X/Y/Z$ kan aanneem.

Al die funksies (A , E , S) wat onder Y gelys word, is monotoon stygend in hierdie proefskrif.

Indien daar byvoorbeeld na die probleem K/A , E/V verwys word, verwys dit na 'n HTP met die vorm soos in (1.1) met al die funksies $f_i(x_i)$ in die doelfunksie wat óf konkaf óf konveks is, en met die lengtes van al die intervalle $[l_i, u_i]$ ewe groot. Dit beteken $u_1 - l_1 = \dots = u_i - l_i = \dots = u_n - l_n$.

1.2 Toepassings

Daar is baie toepassings van nielineêre HTP's. In die volgende afdelings word net 'n paar van die talle interessante toepassings gelys.

1.2.1 Bemarking

Hierdie voorbeeld is uit Lilien en Kotler [18] se boek geneem. Gestel 'n produk moet bemark word. Daar kan aanvaar word dat die produk ál beter sal verkoop met 'n toename in die bemarkingspoging. Die verloop van die verkope van die produk wat bemark word as 'n funksie van die bemarkingspoging, kan in die algemeen een van drie vorms aanneem:

1. 'n *Konvekse stygende funksie*: Dit beteken dat die verkope aanvanklik nie so vinnig styg nie, maar soos die bemarkingspoging toeneem, styg die verkope al hoe vinniger.
2. 'n *Konkawe stygende funksie*: In hierdie geval sal die verkope aan die begin vinnig styg met 'n toename in die bemarkingspoging, maar namate die mark versadig raak, sal die verkope van die produk al hoe stadiger styg met 'n toename in die bemarkingspoging.
3. 'n *S-kromme*: Hierdie kromme is eintlik net 'n samestelling van die voorafgaande twee situasies en miskien die mees realistiese. Die verkope styg aanvanklik stadig met 'n toename in die bemarkingspoging. In die middel styg dit vinniger nadat 'n sekere bemarkingsvlak bereik is en begin dan afplat namate die mark versadig raak.

Indien die bemarkingspoging – dit kan byvoorbeeld met die randbedrag wat aan bemarking bestee word, benader word – van produk i as x_i gedefinieer word en die verkope van produk i gedefinieer word as 'n funksie van x_i , gegee deur $f_i(x_i)$, kan die probleem as

$$\text{maks} \quad \sum_{i=1}^n f_i(x_i)$$

onderhewig aan

$$\sum_{i=1}^n x_i \leq B$$

$$x_i \geq 0 \quad i = 1, 2, 3, \dots, n,$$

gemodelleer word, waar n die aantal produkte is wat bemark moet word en B die begroting wat aan bemarking bestee kan word. Lilien en Kotler [18] bespreek die oplossing van die geval waar al die funksies $f_i(x_i)$ stygend en konkaf is. Hulle stel egter geen oplossingsmetode vir die S-kromme voor nie, behalwe om te sê dat dit met 'n stygende konkawe funksie benader kan word. Oor die maksimering van konvekse funksies word ook geen woord geryp nie.

1.2.2 Chemie

Sekere chemiese produksieprosesse kan ook as 'n HTP gemodelleer word [26]. Heelwat chemiese prosesse wat tydens produksie plaasvind, word gekenmerk deur 'n deurvoertempo – die gemiddelde produksiespoed van die chemiese proses per eenheidstyd – wat eksponensieel oor tyd verval. Hierdie prosesse moet van tyd tot tyd “gediens” word om vars grondstowwe (reagense) by te voeg en afvalmateriaal te verwyder. Die probleem is om die intervalle tussen hierdie dienste te bepaal. Hierdie intervalle bepaal dan weer die kapasiteit van die fasiliteit. Laat z_j die dienstempo wees met j 'n indeks oor al die verskillende take wat in dieselfde fasiliteit voltooi moet word. Die deurvoertempo is proporsioneel tot die oorblywende grondstof en het dus die produksiefunksie

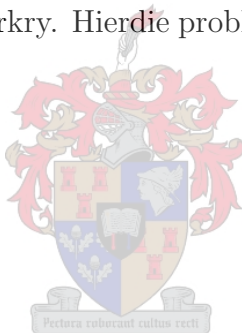
$$a_j z_j (1 - e^{-b_j/z_j}),$$

tot gevolg, waar $a_j, b_j \geq 0$ en konstantes is. Indien $x_j = z_j/b_j$, word die algemene vorm

$$f_i(x_i) = x (1 - e^{-1/x})$$

van die funksies in die doelfunksie verkry. Hierdie probleem kan dan getransformeer word na

$$\text{maks} \quad \sum_{i=1}^n \alpha_i f_i(x_i)$$



onderhewig aan

$$\sum_{i=1}^n \beta_i x_i \leq K$$

$$x_i \geq 0 \quad i = 1, 2, 3, \dots, n,$$

wat dit 'n HTP maak.

1.2.3 Kapasiteitsbepaling in vervaardiging, gesondheidsorg en rekenaarnetwerke

Hierdie baie interessante toepassing van HTP kom uit [7]. Heelwat sisteme kan gemodelleer word as 'n netwerk van toue, waar elke nodus in die netwerk 'n diensfasiliteit is met kliënte wat in toue wag om bedien te word. In die bepaling van die kapasiteit in vervaardiging moet die koste van vervaardiging geminimeer word, onderhewig aan sekere dienstempo's by elke werkstasie met 'n bogrens op die totale geldwaarde van die werk-in-voorraad. In gesondheidsorg is hierdie soortgelyke probleem die een waarin die koste om kapasiteit in stand te hou, geminimeer moet word onderhewig aan die beperking op die verwagte tydsduur van 'n pasiënt in die fasiliteit. In rekenaarnetwerke is die beperking

net die limiet op hoe lank die vertraging van byvoorbeeld e-posboodskappe mag duur. In die algemeen kan die kapasiteitsbepalingsprobleem geformuleer word as

$$\min \sum_{i=1}^n f_i(x_i)$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^n b_i \left(\frac{\alpha_i}{x_i - \alpha_i} \right) &\leq B \\ l_i \leq x_i &\leq u_i && i = 1, 2, 3, \dots, n, \\ x_i &\geq 0 && i = 1, 2, 3, \dots, n, \end{aligned}$$

waar die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$ die koste van kapasiteit by 'n dienstempo van x_i is, α_i die aankomstempo by werkstasie i is, $\frac{\alpha_i}{x_i - \alpha_i}$ die verwagte aantal items (take) wat wag by stasie i is, b_i en B reële getalle groter as nul is.

In vervaardiging is b_i die gemiddelde geldwaarde van die werk-in-voorraad wat wag by stasie i , en B die bogrens op die totale hoeveelheid werk-in-voorraad in die netwerk.

1.2.4 Portefeulje

Selfs die finansiële wêreld spring die toepassingsmoontlikhede van HTP's nie vry nie. Die volgende toepassing word in [7] beskryf. In hierdie geval word die optimale toewysing van hulpbronne aan 'n seleksie van n verskillende beleggings gesoek. Indien die opbrengs van 'n belegging 'n stogastiese veranderlike is, moet die belegger sy besluit maak op grond van beide verwagte opbrengs sowel as die variansie van daardie belegging. Dit is dus verkieslik om 'n belegging met 'n kleiner variansie te maak. Dit sal dus sin maak om 'n doelfunksie te hê wat die verhouding van verwagte opbrengs tot variansie maksimeer. Laat x_i die breuk van die hulpbron wees wat in belegging i gestort word. Laat R_i die verwagte opbrengs van belegging i wees, en σ_{ij} die kovariansie tussen beleggings i en j . Die verwagte opbrengs word dan gegee deur

$$\sum_{i=1}^n R_i x_i$$

en die kovariansie word gegee deur

$$\left(\sum_{i,j} \sigma_{ij} x_i x_j \right)^{\frac{1}{2}}.$$

Die probleem kan dus geformuleer word as

$$\text{maks} \left[\frac{\sum_{i=1}^n R_i x_i}{\left(\sum_{i,j} \sigma_{ij} x_i x_j \right)^{\frac{1}{2}}} \right]$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^n x_i &= 1 \\ x_i &\geq 0, \quad i = 1, 2, 3, \dots, n. \end{aligned}$$

In hierdie geval is die doelfunksie nie skeibaar nie. In die spesiale geval waar die korrelasiekoëffisiënte tussen i en j konstant is, volg vir $0 < \rho < 1$ dat $\sigma_{ii} = \sigma^2$ vir $i = 1, 2, \dots, n$ en $\sigma_{ij} = \rho \sigma_i \sigma_j$, vir $i, j = 1, 2, \dots, n$, $i \neq j$. Deur gebruik te maak van die Kuhn-Tucker-voorwaardes kan die probleem getransformeer word na

$$\text{maks} \left[\sum_{i=1}^n R_i x_i - \frac{1-\rho}{2} \sigma_i^2 x_i^2 \right] - \frac{1}{2} \left(\sum_{i=1}^n \sigma_i x_i \right)^{\frac{1}{2}}$$

onderhewig aan

$$x_i \geq 0 \quad i = 1, 2, 3, \dots, n.$$

Stel $y_i = \sigma_i x_i$ en $-w = \sum_{i=1}^n \sigma_i x_i$. Die probleem kan dan herskryf word as

$$\text{maks} \left[\sum_{i=1}^n \left\{ \frac{(1-\rho)y_i^2}{2} - R_i \frac{y_i}{\sigma_i} \right\} + \frac{w^2}{2} \right]$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^n y_i + w &\geq 0 \\ y_i &\geq 0 \quad i = 1, 2, 3, \dots, n \\ w &\text{teken onbeperk.} \end{aligned}$$

1.2.5 Produksiebeplanning en voorraadbeheer

Dat HTP's toepassings in die logistieke omgewing sou vind, kom nie as 'n verrassing nie. Die volgende toepassing in die produksie-omgewing kom uit die boek van Ibaraki en

Katoh [16]. In sommige produksiemodelle moet die hulpbron (produksie-ure) toegeedeel word aan elke item wat geprosesseer word, sodat die totale koste geminimeer word. Een so 'n model is die bepaling van 'n optimale bestelgrootte of produksielopie. In hierdie geval word die doelfunksie gegee deur die som van die bestelkoste en voorraaddrakoste. Die enkele beperking kan byvoorbeeld 'n beperking op die totale stoorruimte wees. In hierdie geval word die formulering gegee deur

$$\min \sum_{i=1}^n \left[\frac{D_i}{y_i} C_{0i} + \frac{1}{2} C_{hi} y_i \right]$$

onderhewig aan

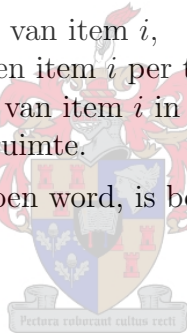
$$\begin{aligned} \sum_{i=1}^n b_i y_i &\leq N \\ y_i &\geq 0 \quad i = 1, 2, 3, \dots, n. \end{aligned}$$

In hierdie formulering is:

- y_i = die bestelgrootte van item i ,
- D_i = die totale vraag na item i ,
- C_{0i} = die koste van een bestelling van item i ,
- C_{hi} = die voorraaddrakoste van een item i per tydseenheid,
- b_i = die ruimte wat een eenheid van item i in beslag neem,
- N = die totale beskikbare stoorruimte.

Indien die transformasie $x_i = b_i y_i$ gedoen word, is bostaande formulering 'n HTP.

1.2.6 Soektogte



'n Veld binne operasionele navorsing wat die afgelope tyd heelwat aandag getrek het by navorsers, is die soeke na optimale soekstrategieë. HTP's het ook hul weg na toepassings in hierdie gebied gevind. Die voorbeeld wat hieronder gegee word, kom ook uit Ibaraki en Katoh [16]. Gestel 'n objek wat by of tussen een van die liggings van 1 tot n is, moet opgespoor word. Die waarskynlikheid p_i dat die objek by punt i is, is bekend. 'n Soekpoging moet van stapel gestuur word om die objek op te spoor. Gestel die voorwaardelike waarskynlikheid dat die objek by punt i is, is $(1 - e^{-\alpha x_i})p_i$, wanneer x_i – die hoeveelheid van die soekpoging – aan punt i toegeedeel word en α 'n positiewe konstante is. Die totale grootte van die soekpoging is N . Die probleem kan dan geformuleer word as

$$\text{maks} \sum_{i=1}^n (1 - e^{-\alpha x_i}) p_i$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^n x_i &= N \\ x_i &\geq 0. \end{aligned}$$

Die probleem kan diskreet of kontinu wees, afhangende daarvan of die grootte van die soekpoging x_i diskreet of kontinu is.

1.2.7 Steekproewe

Statistiek is 'n naverwante vak aan operasionele navorsing en daarom is dit nie verbasend nie dat HTP's ook in hierdie vakrigting toegepas kan word. Een so 'n toepassingsmoontlikheid word deur Bretthauer en Shetty [7] gegee. Beskou die probleem waar 'n populasie se gemiddelde μ beraam moet word met 'n waarde \bar{y} . Een benadering om hierdie probleem op te los, is om die populasie te verdeel in n strata, steekproewe te trek uit elke stratum en dan die populasiegemiddelde te beraam met 'n geweege gemiddelde van die gemiddeldes van die strata. Die probleem om die getal eenhede te bepaal wat uit elke stratum in die steekproef ingesluit moet word, kan as 'n HTP gemodelleer word. In hierdie model sal die veranderlikes x_i die steekproefgrootte in elkeen van die strata wees. Die probleem kan dan geformuleer word met die doel om die variansie in \bar{y} te minimeer onderhewig aan 'n beperking op die koste van die steekproefneming. Die wiskundige formulering word gegee deur

$$\min \sum_{i=1}^n V(\bar{y})$$

onerhewig aan

$$\sum_{i=1}^n b_i x_i \leq b$$

$$l_i \leq x_i \leq u_i$$

$$x_i \geq 0$$



$$i = 1, 2, 3, \dots, n$$

$$i = 1, 2, 3, \dots, n.$$

In hierdie formulering word die volgende notasie gebruik:

- n = aantal strata,
- x_i = steekproefgrootte vir stratum i ,
- l_i = ondergrens op die steekproefgrootte vir stratum i ,
- u_i = bogrens op die steekproefgrootte vir stratum i ,
- N_i = aantal eenhede in stratum i ,
- N = aantal eenhede in die populasie ($N = \sum_{i=1}^n N_i$),
- σ_i = standaardafwyking van stratum i ,
- \bar{y}_i = steekproefgemiddelde vir stratum i ,
- μ = populasiegemiddelde,
- b = beskikbare begroting vir steekproefneming, en
- b_i = koste om die inligting oor een eenheid in stratum i te verkry.

Dit kan aangetoon word dat μ onsydig beraam kan word as die geweege gemiddelde van die gemiddeldes van die steekproewe uit al die strata. Dit wil sê

$$\bar{y} = \frac{1}{N} \sum_{i=1}^n N_i \bar{y}_i.$$

Die variansie van hierdie beraming is

$$V(\bar{y}) = \frac{1}{N^2} \sum_{i=1}^n \frac{N_i^2 \sigma_i^2 (N_i - x_i)}{N_i x_i}.$$

Laat $d_i = (N_i \sigma_i / N)^2$ en $D = \sum_{i=1}^n d_i / N_i$ wees, dan kan die steekproefprobleem hierbo herskryf word as

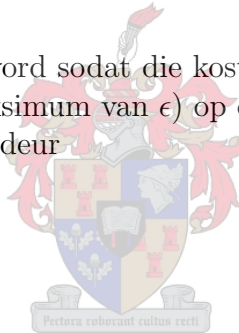
$$\min \sum_{i=1}^n \frac{d_i}{x_i} - D$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^n b_i x_i &\leq b \\ l_i &\leq x_i \leq u_i && i = 1, 2, 3, \dots, n \\ x_i &\geq 0 && i = 1, 2, 3, \dots, n. \end{aligned}$$

Die probleem kan ook geformuleer word sodat die koste van die steekproefneming geminimeer word, met 'n beperking (maksimum van ϵ) op die variansie van die beramer. Die probleem se formulering word gegee deur

$$\min \sum_{i=1}^n f_i(x_i)$$



onderhewig aan

$$\begin{aligned} V(\bar{y}) &\leq \epsilon \\ l_i &\leq x_i \leq u_i && i = 1, 2, 3, \dots, n \\ x_i &\geq 0 && i = 1, 2, 3, \dots, n. \end{aligned}$$

In al die bostaande formuleringe kan geëis word dat x_i heeltallig moet wees. In die geval waar x_i redelik groot is, kan aangetoon word dat die afrondings van die x_i 's ook goeie antwoorde verskaf.

1.2.8 Hulpbronverdeling

In hierdie geval is daar n plekke waar daar 'n vraag na 'n hulpbron bestaan. Die probleem wat opgelos moet word, is wat die optimale verhoudings (verdeling) is waarin hierdie hulpbron na al hierdie punte versprei moet word. Neem die verspreiding van koerante as voorbeeld. Gestel die vraag na koerante by elkeen van die n punte is onseker en 'n lineêre koste word ondervind indien daar 'n onder- of ooraanbod is. Laat $q_j(x_j)$ die verwagte koste by punt j wees indien x_j koerante aan daardie punt toegeken is. Aanvaar die vraag

na koerante by elke punt is kontinu en dat die gemiddelde vraag positief is. Definieer die volgende veranderlikes:

- F_j = kumulatiewe verdelingsfunksie van die vraag by punt j ,
- μ_j = die gemiddeld van F_j ,
- γ_j = eenheidskoste van 'n tekort by punt j ,
- δ_j = eenheidskoste van 'n oorskot by punt j , en
- β_j = beginvoorraad by punt j .

Die verwagte koste by punt j word dan gegee deur

$$\begin{aligned}
 q_j(x_j) &= \int_0^{x_j+\beta_j} (x_j + \beta_j - y) dF_j(y) + \gamma_j \int_{x_j+\beta_j}^{\infty} (y - x_j - \beta_j) dF_j(y) \\
 &= -\delta \int_0^{x_j+\beta_j} y dF_j(y) + \gamma \int_0^{\infty} y dF_j(y) - \gamma \int_0^{x_j+\beta_j} y dF_j(y) \\
 &\quad + \delta_j(x_j + \beta_j) \int_0^{x_j+\beta_j} dF_j(y) - \gamma_j(x_j + \beta_j) \int_0^{\infty} dF_j(y) \\
 &\quad + \gamma_j(x_j + \beta_j) \int_0^{x_j+\beta_j} dF_j(y) \\
 &= -(\gamma_j + \delta_j) \int_0^{x_j+\beta_j} y dF_j(y) + \gamma_j \mu_j + (\gamma_j + \delta_j)(x_j + \beta_j) \int_0^{x_j+\beta_j} dF_j(y) \\
 &= \gamma_j(\mu_j - x_j - \beta_j) + (\gamma_j + \delta_j) \int_0^{x_j+\beta_j} F_j(y) dy.
 \end{aligned}$$

Die laaste stap volg uit die feit dat

$$\mu_j = \int_0^{\infty} y dF_j(y)$$

en

$$\int_0^{\infty} dF_j(y) = 1.$$

Dit kan aangetoon word dat alle $q_j(x_j)$ konveks is. Die resulterende HTP word gegee deur

$$\text{maks} \quad \sum_{j=1}^n q_j(x_j)$$

onderhewig aan

$$\begin{aligned}
 \sum_{j=1}^n x_j &\leq N \\
 x_j &\geq 0 \quad \quad \quad j = 1, 2, 3, \dots, n.
 \end{aligned}$$

Hierdie toepassing van 'n HTP kom ook uit die boek van Ibaraki en Katoh [16].

1.3 Doelwitte van die proefskrif

Die ontwikkeling van algoritmes wat 'n globale optimum lewer vir die HTP in (1.1), word as 'n oorkoepelende doelwit gestel. In enkele gevalle word goeie heuristiese algoritmes ook gegee. Hierdie doelwit kan in die volgende subpunte verfynd word:

1. Verskaf 'n kort oorsig oor watter werk en metodes in die literatuur oor hierdie onderwerp bestaan. Daar bestaan reeds baie goeie oorsigartikels oor hierdie probleem [5, 16, 31, 41].
2. Gee 'n oorsig oor die bestaande teorie wat as fondament van die oplossing van nielineêre HTP's dien.
3. Ontwikkel algoritmes om die HTP waarvan die doelfunksie uit skeibare, monotoon stygende konvekse funksies bestaan, op te los. Onderzoek die oplossingstye van hierdie algoritmes soos volg:
 - (a) ontwikkel eers algoritmes vir die spesiale (eenvoudiger) geval waar al die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$ oor vaste intervalle gemaksimeer word; en
 - (b) ontwikkel dan algoritmes vir die meer algemene geval waar die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$ oor wisselende intervalle gemaksimeer word.
4. Ontwikkel algoritmes om die HTP waarvan die doelfunksie uit skeibare, monotoon stygende konvekse en konkawe funksies bestaan, te optimeer. Onderzoek die kompleksiteit van hierdie algoritmes.
5. Vergelyk verskillende algoritmes met mekaar en maak aanbevelings oor die bruikbaarheid daarvan.
6. Verskaf 'n kort samevatting oor wat bereik is en gee idees oor moontlikhede vir verdere ondersoek.

1.4 Proefskrifuitleg

Oorsigtelik is die uitleg van die proefskrif soos volg: Hoofstuk 1 bevat 'n kort inleiding, agtergrond en toepassings van HTP's. In Hoofstuk 2 word 'n oorsig oor bestaande navorsing asook bestaande teorie verskaf. In Hoofstuk 3 word oplossings vir die spesiale geval van $K/E/V$ voorgestel. In Hoofstukke 4 en 5 word die oplossingsmetodes veralgemeen na $K/E/G$. Oplossingsmetodes vir die veralgemening na $K/E, A/G$ word dan in Hoofstukke 6 en 7 aangebied. In Hoofstuk 8 word 'n samevatting en idees vir verdere studie aangebied.

Die uitleg van die proefskrif word in meer besonderhede hieronder uiteengesit.

Die eerste hoofstuk van die proefskrif gee 'n kort beskrywing van die probleem wat in die proefskrif onder die loep geneem gaan word. Verskeie toepassings van HTP's in 'n diverse verskeidenheid van vakgebiede word ook gegee.

In Hoofstuk 2 volg 'n kort oorsig oor bestaande literatuur. Hierdie oorsig is inderdaad kort aangesien daar nie veel literatuur bestaan oor $K/E/G$ - en $K/E, A/G$ -HTP's nie. Daar bestaan klassieke metodes om nielineêre optimeringsprobleme in die algemeen op te los, soos lineêre omsluiting saam met vertak-en-begrens-metodes. Dit is ook bekend dat dinamiese programmering gebruik kan word om algemene nielineêre knapsakprobleme op te los. Behalwe vir hierdie klassieke, lomp benaderings, kon slegs vier publikasies opgespoor word wat spesifiek $K/E/G$ - (of die transformasie, soos $N/A/G$ -HTP's, daarvan) en $K/E, A/G$ -HTP's probeer oplos. Twee van hierdie vier publikasies verskaf net heuristiese benaderings. Die ander twee publikasie wat wel eksakte oplossings soek, beskou slegs kwadratiese funksies in die doelfunksie en heeltallige oplossings onderskeidelik.

Die nodige bestaande teoretiese onderbou vir die latere algoritmes word ook in Hoofstuk 2 aangebied. Dit is hoofsaaklik (minder bekende) teorie oor nielineêre optimering. In hierdie hoofstuk word klassieke nodige en voldoende voorwaardes vir nielineêre optimering beskou. Twee algoritmes wat van lineêre omsluiting gebruik maak, word ook beskou. Enkele ander teoretiese konsepte wat later gebruik word, soos Taylor se stelling en die begrip van die kromming van 'n funksie, word ook verskaf.

In Hoofstuk 3 word 'n subversameling van HTP's, naamlik $K/E/V$ -HTP's beskou. Twee nuwe algoritmes word voorgestel wat hierdie subversameling van konvekse funksies kan optimeer. Hierdie algoritmes word die breukalgoritme en die hellingalgoritme genoem. Althoe hierdie algoritmes gebruik die feit dat die funksie wat 'n toedeling tussen sy bo- en ondergrens kry (waarvan daar hoogstens een kan wees) altyd 'n vaste toedeling sal kry.

In Hoofstuk 4 word soekalgoritmes aangebied wat $K/E/G$ -HTP's kan optimeer. Die blaarsoek- en taksoekalgoritmes berus op 'n binêre soektog om die optimale oplossing te bepaal. Hierdie algoritmes gebruik ook die feit dat hoogstens een funksie 'n toedeling tussen sy bo- en ondergrens kan kry. Die algoritmes berus daarop dat takke en/of blare in die binêre boom waarin daar nie 'n toelaatbare oplossing kan voorkom nie, vroegtydig opgespoor en geëlimineer word om sodoende die soektog te verklein. Hierdie algoritmes lewer egter nie goeie resultate nie.

'n Heuristiek en drie algoritmes, wat almal op vertak-en-begrensing berus, word in Hoofstuk 5 voorgestel.

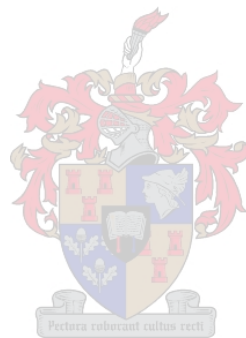
Van al die algoritmes vir $K/E/G$ -HTP's wat in hierdie proefskrif aangebied word, is hierdie versameling van algoritmes verreweg die effektiëste om $K/E/G$ -HTP's te optimeer. Die heuristiek wat aangebied word, los eintlik net 'n subversameling van die vertak-en-begrens-boom op – juis daarom lewer dit net soms die optimale oplossing. Die eerste algoritme stel 'n vertak-en-begrens-metode voor wat met elke iterasie 'n subversameling van probleme in plaas van 'n enkele subprobleem in die vertak-en-begrens-boom oplos. Die tweede algoritme gebruik ook vertak-en-begrensing maar sny ook (indien moontlik) tydens elke vertakking gedeeltes van die toelaatbare gebied af. Die dele van die toelaatbare gebied wat afgesny word, is gedeeltes waarvoor die optimale oplossing reeds bekend is. Die derde algoritme kombineer die boonste twee algoritmes in die sin dat dit subversamelings van probleme oplos en die toelaatbare gebied kleiner sny tydens iterasies (vertakkings).

In Hoofstuk 6 word daar gepoog om 'n kontinue benadering in plaas van 'n vertak-en-begrens-benadering te volg. In die eerste deel van hierdie hoofstuk word die eenvoudiger geval van slegs een konvekse en een konkawe funksie beskou. Dit is moontlik om die $K/E, A/G$ -HTP te reduseer na 'n HTP met slegs twee versamelings van funksies in die doelfunksie – een groep konkav, wat as een konkawe funksie beskou kan word, en die ander groep konveks wat almal net waardes by hul bo- of ondergrense kan aanneem. Hierdie situasie word ook in Hoofstuk 6 ondersoek. In hierdie geval word klassieke optimerings-tegnieke tesame met die spesiale struktuur vir hierdie probleem en funksies gebruik in 'n poging om die kontinue optimale oplossing vir 'n $K/E, A/G$ -HTP te bepaal. Hierdie benadering blyk egter nie effektief te wees nie.

Die probleem waarin daar meer as een konkawe en konvekse funksies in die doelfunksie voorkom, kry in Hoofstuk 7 aandag. In hierdie hoofstuk word daar ook van vertak-en-begrens-algoritmes gebruik gemaak om optimale oplossings vir die $K/E, A/G$ -HTP te bepaal. Ongeveer dieselfde logika as wat in Hoofstuk 5 vir die goeie resultate verantwoordelik is, word veralgemeen om die versameling konkawe funksies ook te akkommodeer. Ongelukkig is die verbetering in oplossingstye in hierdie geval nie so drasties as in Hoofstuk 5 waar slegs konvekse funksies beskou word nie.

'n Samevatting van die resultate van hierdie studie word in Hoofstuk 8 gegee. Idees vir verdere studie word ook in hierdie hoofstuk genoem.

Laastens word die rekenaarkode, geprogrammeer in *Visual Basic for Applications (VBA)*, vir al die algoritmes wat in hierdie proefskrif aangebied word, verskaf. Dit is beskikbaar op die kompakskyf wat op die agterblad ingebind is. Dit is belangrik om daarop te let dat dit nie een van die doelwitte van hierdie proefskrif is om 'n gebruikersvriendelike pakket te skep om die betrokke HTP's op te los nie. Die programme wat op die kompakskyf verskaf word, is dus glad nie uit 'n gebruikersoogpunt geprogrammeer sodat dit maklik deur 'n derde party gebruik kan word nie. Die programmatuur bevat byvoorbeeld geen kode om te toets dat geldige data ingevoer word nie en geen gebruikershandleiding word verskaf nie. Die rede waarom die kode verskaf word, is gewoon net om dit beskikbaar te stel aan 'n persoon wat graag wil sien hoe die algoritmes in rekenaarkode geïmplementeer kan word.



Hoofstuk 2

Literatuur

Heelwat publikasies oor die hulpbrontoekeningsprobleem (HTP) het al die lig gesien. Die meeste publikasies fokus egter op die probleme waarin die minimering van konvekse of die maksimering van konkawe funksies beskou word. In hierdie proefskrif word gefokus op HTP's waarin die som van konvekse funksies gemaksimeer word, asook op HTP's waarin die som van 'n kombinasie van konvekse en konkawe funksies gemaksimeer word.

Daar is al baie teoretiese resultate ontwikkel en gepubliseer in verband met nielineêre optimering en kalkulus. Om hierdie proefskrif self omvattend te maak en nie die vloei van argumente in komende hoofstukke te onderbreek deur na reeds bekende (en minder bekende) teorie te verwys nie, word 'n kort opsomming in §2.7 tot §2.11 van bestaande teorie gegee. Verwysings na waar meer inligting oor hierdie onderwerpe verkry kan word, word ook verskaf. Dit maak dit ook moontlik om die proefskrif so aan te bied dat bestaande werk en nuwe werk in aparte hoofstukke aangebied kan word om enige misverstand oor wat presies die bydrae van die proefskrif is, uit die weg te ruim.

2.1 Maksimering van konvekse (minimering van konkawe) doelfunksies

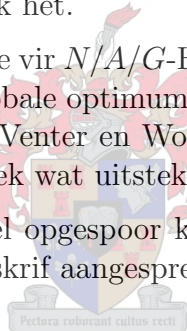
Tot op hede het hierdie probleem baie min (feitlik geen) aandag in die literatuur gekry nie. Die paar pogings in die literatuur om hierdie probleem op te los, het die probleem in die vorm van die minimering van konkawe funksies beskou. Die jongste artikels waarin verwys word na hierdie tipe HTP's is twee artikels deur dieselfde outeurs, naamlik Bretthauer en Shetty [5, 7]. In hul oorsigartikel oor die nielineêre HTP gee hulle 'n baie kort bespreking van hierdie geval. Twee belangrike punte word egter in hierdie enkele paragraaf wat hulle daaraan afstaan gemaak, naamlik dat: (1) die bestaande manier om hierdie probleem op te los, lineêre omsluiting saam met vertak-en-begrensing is; en (2) dat baie min navorsing oor hierdie geval gepubliseer is. In e-poskommunikasie met Michael Patriksson [30] op grond van sy monumentale verslag oor $N/E/G$ - en $K/A/G$ -HTP's, wat ± 300 verwysings bevat [29] en vir publikasie voorgelê is, bevestig dat daar inderdaad baie min oor $K/E/G$ - en $K/E, A/G$ -HTP's gepubliseer is.

Die eerste artikel deur Bretthauer en Shetty [5] stel 'n oplossingsmetode vir die kontinue nielineêre HTP voor indien 'n konvekse doelfunksie geminimeer word oor 'n konvekse beperking. Die metode word ook uitgebrei om 'n paar spesiale vorme, soos die kwadratiese HTP, op te los. In 'n afdeling oor die niekonvekse geval, *i.e.* minimering van niekonvekse funksies, verduidelik hulle kortliks hoe om 'n lineêre omsluiting vir die niekonvekse geval te skep. Hulle stel dan voor dat die lineêre omsluiting met hulle algoritme opgelos word. Gewone vertak-en-begrensing kan dan gebruik word om die globale optimale oplossing te vind. Bretthauer en Shetty se algoritme is geskryf om die nielineêre probleem op te los en die omsluiting, soos hulle dit definieer, beteken dat die omsluiting met reguit lyne gedoen word wat dus 'n liniëring van die probleem beteken. Hierdie lineêre HTP kan dan meer effektief opgelos word met algoritmes wat die lineêre HTP oplos. Aan die einde van die paragraaf verwys hulle die leser na die werk van Horst en Tuy [15] oor hoe vertak-en-begrens-metodes werk.

Die tweede artikel deur Bretthauer en Shetty [7] is 'n oorsigartikel oor oplossingsmetodes vir die nielineêre knapsakprobleem. Aan die einde van hierdie artikel sluit hulle weer 'n kort paragraaf oor die niekonvekse probleem in wat feitlik verbatim herhaal wat in die vorige artikel staan. Aan die begin van hierdie paragraaf word wel vermeld dat Moré en Vavasis [27] hierdie probleem aangepak het.

Venter [40] stel ook 'n oplossingsmetode vir $N/A/G$ -HTP's voor. Aanvanklik word hierdie algoritme aangebied as een wat die globale optimum vir $N/A/G$ -HTP's bepaal, maar dit word dan in 'n latere publikasie deur Venter en Wolvaardt [41] versag tot 'n heuristiek. Dit is inderdaad 'n baie goeie heuristiek wat uitstekende resultate lewer.

Hieronder volg die publikasies wat wel opgespoor kon word en min of meer verwant is aan die probleem wat in hierdie proefskrif aangespreek word.



2.1.1 Luss en Gupta

Alhoewel Luss en Gupta [22] nie $K/E/G$ - of $N/A/G$ -HTP's beskou nie, word hulle hier genoem omdat hulle die vier families van funksies wat gebruik word om die toetsprobleme op te stel, voorgestel het. Hulle stel 'n rekursiewe algoritme voor wat 'n $N/E/G$ -HTP wat slegs uit hierdie funksies bestaan, kan oplos.

2.1.2 Moré en Vavasis

Moré en Vavasis [27] stel in hulle artikel 'n oplosmetode voor vir $N/A/G$ -HTP's voor wat presies dieselfde probleem as $K/E/G$ is. Hierdie oplossingsmetode is slegs 'n heuristiek. Die werking daarvan berus rofweg daarop dat die funksies een vir een in volgorde van laagste tot hoogste beginhelling, dit wil sê $f'_i(l_i)$, by hul bogrense vasgestel word totdat die funksie wat 'n toedeling tussen sy onder- en bogrens kry, bepaal is. Die oplossingsmetode is baie vinnig, maar lewer nie goeie resultate vir die probleme waar die funksies kruis nie. Venter [40] gee 'n volledige bespreking van hierdie metode asook die tekortkominge daarvan.

2.1.3 Venter en Wolvaardt

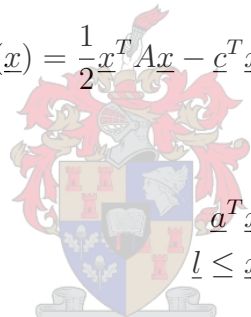
Venter en Wolvaardt [41] stel 'n heuristiek voor vir die minimering van die konkawe knapsakprobleem. Die algemene probleem wat hulle oplos, het presies dieselfde algemene formulering as die probleem wat Moré en Vavasis [27] aangepak het. Die heuristiek berus op 'n iteratiewe soektog na die (hoogstens) een funksie wat 'n toedeling tussen sy bo- en ondergrens moet kry. Een van die sterkpunte van hierdie algoritme is dat dit selfs tabelfunksies kan hanteer. Venter en Wolvaardt se heuristiek lewer egter beduidend beter oplossings as dié van Moré en Vavasis.

2.1.4 Dai en Fletcher

Dai en Fletcher [9] bied 'n eksakte algoritme vir 'n niekonvekse knapsakprobleem aan, maar net kwadratiese funksies in die doelfunksie word beskou. Hulle algoritme berus op 'n raaklynbenadering. Dit verbeter op die bestaande een deur 'n verbetering in stapgrootte en 'n aanpasbare niemonotoon-lynsoektog te inkorporeer. Die algemene vorm van die probleem wat hulle oplos, word gegee deur

$$\min f(\underline{x}) = \frac{1}{2} \underline{x}^T A \underline{x} - \underline{c}^T \underline{x}$$

onderhewig aan



$$\begin{aligned} \underline{a}^T \underline{x} &= b \\ \underline{l} &\leq \underline{x} \leq \underline{u}, \end{aligned}$$

waar $A \in \mathbb{R}^{n \times n}$ en simmetries is; $\underline{a}, \underline{c}, \underline{l} \geq \underline{0}$ en \underline{u} vektore in \mathbb{R}^n is en b 'n skalaar is.

2.1.5 Sun, Wang en Li

Sun, Wang en Li [36] beskou heeltallige oplossings vir die minimering van nietoenemende konkawe doelfunksies. Hulle stel 'n effektiewe en eksakte algoritme voor vir heeltallige oplossings van hierdie probleem. Die algoritme bestaan uit 'n iteratiewe proses tussen die bepaling van onder- en bogrense deur middel van lineêre onderskattings van die doelfunksies en deur snitte en partisies van die toelaatbare gebied te maak. Die ondergens word iteratief verbeter met behulp van hierdie snitte en partisies van die toelaatbare gebied. Hierdie iteratiewe verbetering konvergeer na die optimale antwoord in 'n eindige aantal stappe. Sun *et al.* vergelyk hulle algoritme se oplossingstye met dié van dinamiese programmering en kom tot die gevolgtrekking dat hulle algoritme beter resultate lewer. In hierdie resultate het hulle slegs die som van derdegraadse polinome as doelfunksie beskou. Die algemene vorm van die eksperimentele probleem was dus

$$\min f(\underline{x}) = \sum_{j=1}^n (-c_j x_j^3 - d_j x_j^2 - e_j x_j)$$

onderhewig aan

$$g(\underline{x}) = \sum_{j=1}^n b_j x_j \leq b$$

$$\begin{aligned} x_j &\geq l_j, & j &= 1, 2, \dots, n, \\ x_j &\leq u_j, & j &= 1, 2, \dots, n, \\ x_j &\geq 0, \text{ heeltallig} & j &= 1, 2, \dots, n. \end{aligned}$$

Die konstantes is lukraak gegeneer in die volgende intervale: $c \in [0, 1]$, $d \in [1, 10]$, $d \in [1, 20]$ en $b \in [1, 40]$. Alle $l_j = 1$ en alle $u_j = 5$ en

$$b = \sum_{j=1}^n b_j l_j + 0.5 \left[\sum_{j=1}^n b_j (u_j - l_j) \right].$$

2.2 Maksimering van konkawe (minimering van konvekse) doelfunksies

'n Hele aantal algoritmes, sommige vir spesiale gevalle, is al ontwikkel om hierdie formulering op te los. Die meeste van hierdie algoritmes gebruik die Kuhn-Tucker-voorwaardes in iteratiewe algoritmes. Vir 'n goeie oorsig oor algoritmes om hierdie geval op te los, kan [5, 6, 7, 16] byvoorbeeld gebruik word. Die baie volledige dokument oor die nielineêre HTP van Patriksson [29] kan ook gebruik word vir verwysings na algoritmes vir hierdie geval.

2.3 Gemengde doelfunksies

Slegs een artikel kon opgespoor word waarin 'n mengsel van konvekse en konkawe doelfunksies gebruik word. Vavasis [39] beskou in hierdie artikel die niekonvekse kwadratiese HTP en bepaal slegs plaaslike optima. Hierdie artikel is 'n uitbreiding van die algoritme deur Moré en Vavasis [27].

2.4 Tipes funksies in die doelfunksie

In hierdie studie word twee tipes funksies beskou, naamlik monotoon stygende konvekse en monotoon stygende konkawe funksies. Al hierdie probleme word so transformeer dat $l_i = 0$ en $f_i(0) = 0$.

2.4.1 Stygende konvekse funksies

Die versameling van konvekse funksies, naamlik

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

word algemeen [16, 40] in die literatuur gebruik om nielineêre HTP's te skep. Die eerste skrywers wat hierdie funksies voorgestel het, was Luss en Gupta [22]. Die konstantes s_i , m_i en c_i is positiewe parameters waarmee die vorm van die funksie gestel kan word. Hierdie stel funksies kan nie sonder meer in hierdie proefskrif gebruik word nie omdat hulle nie almal monotoon stygend is nie. Hierdie versameling konvekse funksies is dus onderskeidelik aangepas na

$$f_i(x_i) = -s_i + s_i e^{m_i x_i}, \quad (2.1)$$

$$f_i(x_i) = m_i x_i^2 + s_i x_i, \quad (2.2)$$

$$f_i(x_i) = \frac{s_i(u_i + c_i)}{u_i + m_i} - \frac{s_i[(u_i - x_i) + c_i]}{[(u_i - x_i) + m_i]}, \quad m_i > c_i \text{ en} \quad (2.3)$$

$$f_i(x_i) = s_i \ln(1 + m_i u_i) - s_i \ln(1 + m_i(u_i - x_i)) \quad (2.4)$$

om al vier die families van funksies ook monotoon stygend en positief oor die interval (l_i, u_i) te kry. Alle parameters (c_i , m_i en s_i) moet steeds positief wees.

Hierdie konvekse funksies, vergelyking (2.1) tot (2.4), se eerste en tweede afgeleides word in van die latere algoritmes benodig en word dus volledigheidshalwe ook hier gegee. Die eerste afgeleides word gegee deur

$$\begin{aligned} f'_i(x_i) &= s_i m_i e^{m_i x_i}, \\ f'_i(x_i) &= 2m_i x_i + s_i, \\ f'_i(x_i) &= -\frac{s_i[c_i - m_i]}{[(u_i - x_i) + m_i]^2}, \quad m_i > c_i \text{ en} \\ f'_i(x_i) &= \frac{s_i m_i}{[1 + m_i(u_i - x_i)]}, \end{aligned}$$

terwyl die tweede afgeleides gegee word deur

$$\begin{aligned} f''_i(x_i) &= s_i m_i^2 e^{m_i x_i}, \\ f''_i(x_i) &= 2m_i, \\ f''_i(x_i) &= -\frac{2s_i[c_i - m_i]}{[(u_i - x_i) + m_i]^3}, \quad m_i > c_i \text{ en} \\ f''_i(x_i) &= \frac{s_i m_i^2}{[1 + m_i(u_i - x_i)]^2}. \end{aligned}$$

In van die algoritmes wat later gebruik word, is die inverse funksie van die eerste afgeleides nodig, dit wil sê dit het as invoer 'n helling en as afvoer die x -waarde waar die funksie hierdie helling het. Die inverse funksies word gegee deur

$$f_i'^{-1}(h_i) = \frac{\ln\left(\frac{h_i}{s_i m_i}\right)}{m_i},$$

$$f_i'^{-1}(h_i) = \frac{h_i - s_i}{2m_i},$$

$$f_i'^{-1}(h_i) = -\sqrt{-\frac{s_i(c_i - m_i)}{h_i}} + m_i + u_i, \quad m_i > c_i \text{ en}$$

$$f_i'^{-1}(h_i) = -\frac{\frac{s_i m_i}{h_i} - 1}{m_i} + u_i.$$

2.4.2 Stygende konkawe funksies

Die versameling van konkawe funksies

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

volg direk uit die konvekse funksies van Luss en Gupta [22] hierbo. Die konstantes s_i , m_i en c_i is positiewe parameters waarmee die vorm van die funksie verstel kan word. Ook hierdie funksies kan aangepas word na onderskeidelik

$$f_i(x_i) = s_i (1 - e^{-m_i x_i}), \quad (2.5)$$

$$f_i(x_i) = s_i u_i + m_i u_i^2 - s_i(u_i - x_i) - m_i(u_i - x_i)^2, \quad (2.6)$$

$$f_i(x_i) = -\frac{s_i c_i}{m_i} + \frac{s_i(x_i + c_i)}{(x_i + m_i)}, \quad m_i > c_i \text{ en} \quad (2.7)$$

$$f_i(x_i) = s_i \ln(1 + m_i x_i), \quad (2.8)$$

sodat al die funksies konkav, monotoon stygend en positief oor die interval $[l_i, u_i]$ is. Die vereiste dat $c_i, m_i, s_i > 0$ geld steeds.

Hierdie konkawe funksies, vergelykings (2.5) tot (2.8), se eerste en tweede afgeleides word in van die latere algoritmes benodig en word dus volledigheidshalwe ook hier gegee. Die

eerste afgeleides word gegee deur

$$\begin{aligned} f'_i(x_i) &= s_i m_i e^{-m_i x_i}, \\ f'_i(x_i) &= s_i + 2m_i u_i - 2m_i x_i, \\ f'_i(x_i) &= \frac{s_i(m_i - c_i)}{(x_i + m_i)^2}, \quad m_i > c_i \text{ en} \\ f'_i(x_i) &= \frac{s_i m_i}{(1 + m_i x_i)}, \end{aligned}$$

terwyl die tweede afgeleides gegee word deur

$$\begin{aligned} f''_i(x_i) &= -s_i m_i^2 e^{-m_i x_i}, \\ f''_i(x_i) &= -2m_i, \\ f''_i(x_i) &= \frac{-2s_i(m_i - c_i)}{(x_i + m_i)^3}, \quad m_i > c_i \text{ en} \\ f''_i(x_i) &= -\frac{s_i m_i^2}{(1 + m_i x_i)^2}. \end{aligned}$$

Die inverse funksies word in hierdie geval gegee deur

$$\begin{aligned} f_i^{-1}(h_i) &= -\frac{\ln\left(\frac{h_i}{s_i m_i}\right)}{m_i}, \\ f_i^{-1}(h_i) &= -\frac{h_i - s_i - 2m_i u_i}{2m_i}, \\ f_i^{-1}(h_i) &= \sqrt{\frac{s_i(m_i - c_i)}{h_i}} - m_i, \quad m_i > c_i \text{ en} \\ f_i^{-1}(h_i) &= \frac{\frac{s_i m_i}{h_i} - 1}{m_i}. \end{aligned}$$

2.5 Dinamiese programmering as oplossingsmetode

Venter [40] sowel as Sun *et al.* [36] gebruik dinamiese programmering as maatstaf om hul oplossingsmetodes teen te toets. Dinamiese programmering het die voordeel dat dit maklik programmeerbaar is, maar daar is hoofsaaklik drie hoofredes waarom dinamiese programmering nie 'n bevredigende oplossingsmetode is nie.

1. Dinamiese programmering se berekeningskompleksiteit neem toe met die toename in die hulpbron, sowel as met 'n toename in die intervallengte van die veranderlikes.
2. Dinamiese programmering bepaal slegs die beste heeltallige oplossing. Die staplengte van dinamiese programmering kan wel aangepas word om hierdie probleem gedeeltelik te oorkom, maar dit kom met 'n groot toename in berekeningskompleksiteit.

3. Dinamiese programmering se oplossingstyd is aansienlik stadiger as dié van vertak-en-begrens-algoritmes.

Vir volledigheid is hierdie oplossingsmetode ook geprogrammeer en is die nuwe algoritmes se kompleksiteit daarteen gemeet, maar dit is bloot van akademiese waarde aangesien die vertak-en-begrens-metodes dinamiese programmering ver uitpresteer.

Die werking van dinamiese programmering word in enige standaard handboek oor operasionele navorsing (sien byvoorbeeld [13, 17, 37, 44]) behandel en daar gaan dus nie hier verder ruimte daaraan afgestaan word nie.

2.6 Vertak-en-begrens-metodes

In hierdie afdeling word 'n aantal maniere van hoe lineêre omsluiting saam met vertak-en-begrensing gebruik kan word, beskou.

2.6.1 Gewone vertak-en-begrens-algoritme

Die algoritmes wat in hierdie afdeling gegee word, is verwerkings (of aanpassings) van bestaande vertak-en-begrens-metodes om $K/E/G$ -HTP's op te los.

Die een eienskap wat goed benut kan word indien die probleem gelinieer word, is dat indien die funksies $f_i(x_i)$ vir $i = 1, 2, \dots, n$ in die doelfunksie met reguitlyne (tussen enige twee punte) benader word, die lyn altyd 'n groter funksiewaarde as die werklike funksie sal hê. Dit beteken dat die begrensing op grond van die liniëring se oplossing gedoen kan word omdat die doelfunksiewaarde, indien van die liniëring na die werklike doelfunksie beweeg word, altyd sal verminder. Anders gestel, is daar geen beter werklike oplossing as die lineêre omsluiting s'n nie omdat die funksiewaarde altyd kleiner as die omsluiting is.

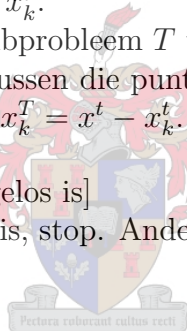
Algoritme 2.1 (Vertak-en-begrens)

Bereken die optimale toewysing van hulpbronne aan 'n aantal konvekse funksies oor wisselende intervalle deur van lineêre omsluiting en vertak-en-begrensing gebruik te maak.

Gestel T is die totale aantal probleme wat in die algoritme opgelos moet word. Gestel verder dat H die totale beskikbare hulpbron is. Die boskrif t dui die onderskeie groothede vir die t -de probleem aan.

1. [Inisialiseer en stel die beginprobleem (omsluiting) op]
Stel T en t gelyk aan 1. Vervang elke funksie $f_i(x_i)$ in die doelfunksie met 'n lyn wat die punte $(l_i, f_i(l_i)) = (0, 0)$ en $(u_i, f_i(u_i))$ vir alle $i = 1, 2, \dots, n$ verbind. Sodoende word probleem (1.1) gelinieer. Stel die grens G (die beste oplossing tot op hede) gelyk aan nul.
2. [Los die volgende subprobleem op]
Los die t -de lineêre HTP, naamlik $HTP(t)$ op.

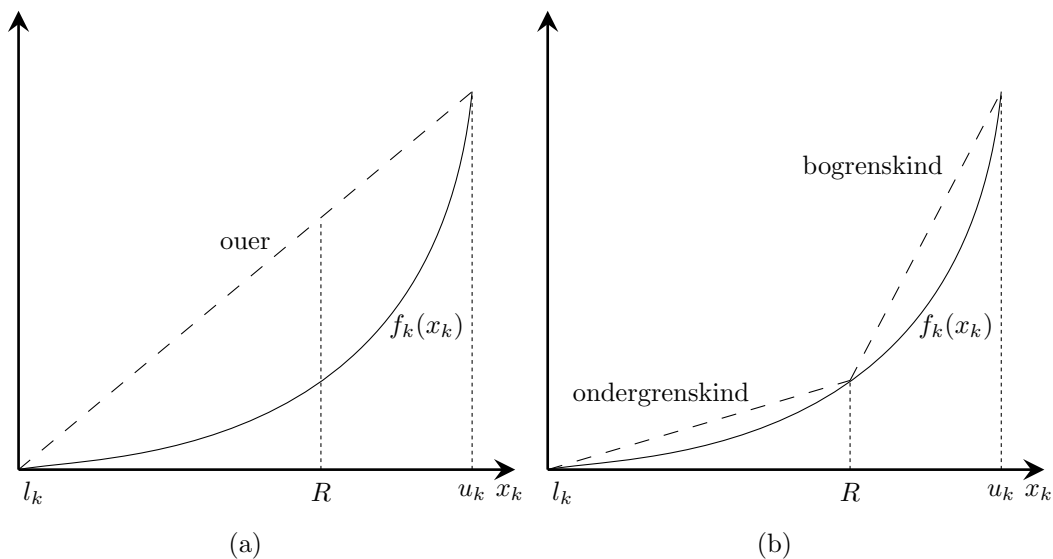
3. [Doen die begrening]
Toets of die doelfunksiewaarde van $HTP(t)$, naamlik $D(t)$, groter/kleiner as die bestaande grens (G) is:
- (a) [$D(t) > G$] Stel die oplossing van $HTP(t)$ terug in die oorspronklike probleem en verkry so die werklike doelfunksiewaarde $W(t)$. Indien hierdie waarde groter as die grens is, werk die grens by met hierdie waarde, $G = W(t)$. Indien geen funksie 'n toedeling tussen sy bo- of ondergrens gekry het nie, gaan na Stap 5. Anders is daar 'n funksie $f_k(x_k)$ wat 'n toedeling tussen sy bo- en ondergrens gekry het. Gaan na Stap 4.
- (b) [$D(t) \leq G$] Gaan na Stap 5.
4. [Voeg die twee nuwe subprobleme by indien die oplossing beter as die bestaande beste oplossing is.]
Vermeerder T met 2
Skep subprobleem $T - 1$: Skep subprobleem $T - 1$ uit probleem t . Vervang funksie k se lyn in probleem $T - 1$ met die lyn tussen die punte $(l_k^{T-1}, f_k(l_k^{T-1}))$ en $(x_k^{T-1}, f_k(x_k^{T-1}))$, en laat $u_k^{T-1} = x_k^t$.
Skep subprobleem T : Skep subprobleem T uit probleem t . Vervang funksie k se lyn in probleem T met die lyn tussen die punte $(x_k^t, f_k(x_k^t))$ en $(u_k^t, f_k(u_k^t))$. Maak die nulpunt transformasie sodat $x_k^T = x^t - x_k^t$.
5. [Toets of al die subprobleme opgelos is]
Indien alle subprobleme opgelos is, stop. Andersins vermeerder t met 1 en gaan na Stap 2.



Algoritme 2.1 verdeel die toelaatbare gebied tydens elke iterasie effektief in twee deur 'n veranderlike se interval in twee te verdeel. Beskou Figuur 2.1: Die funksie, $f_i(x_i)$, word in die Figuur 2.1 (a) en (b) getoon. In Figuur 2.1 (a) word die ouer getoon. Die ouer se doelfunksie word gegee deur lyne wat die punte $(l_i, f_i(l_i))$ en $(u_i, f_i(u_i))$ (vir alle $i = 1, 2, \dots, n$) verbind. Die veranderlikes, beperking en grense op die veranderlikes in die ouer is dieselfde as vir die ekwivalente nielineêre probleem. Gestel die optimale oplossing van die ouer is so dat $x_i = R$, met $R \in (l_i, u_i)$. Die twee subprobleme (kinders) wat dan geskep word, word in Figuur 2.1 (b) gegee.

2.6.2 Lineêre omsluiting

Daar is verskeie metodes om omsluitings te doen. Hierdie omsluitings kan byvoorbeeld met nielineêre funksies gedoen word, maar dit is gewoonlik die maklikste om dit met reguitlyne te doen. Hierdie lineêre omsluiting kan dan vinnig opgelos word met konvensionele lineêre programmering-(LP)-metodes en/of -sagteware. Indien die lineêre omsluiting se optimale oplossing presies ooreenstem met die oorspronklike probleem s'n (dit wil sê die lineêre omsluiting se optimale oplossing stem ooreen met punte waar die lyne aan die



Figuur 2.1: Die vertakking van 'n ouer in twee kinders, naamlik die ondergrenskind en die bogrenskind.

oorspronklike funksies raak), sal die optimale oplossing ook optimaal wees vir die oorspronklike probleem. Indien dit nie ooreenstem nie, kan nog lyne iteratief bygevoeg word sodat die nuwe lyne op die oplossing aan die oorspronklike funksies raak.

Die omsluiting kan op twee maniere aangepak word. Hierdie twee maniere word onderskeidelik in § 2.6.2.1 en § 2.6.2.2 weergegee.

2.6.2.1 Omsluit slegs die funksie in versameling \mathcal{K}

Die eerste manier is deur die nodige voorwaardes in Stelling 3.1 te gebruik wat aandui dat slegs een funksie 'n tob-funksie kan wees. Hierdie benadering beteken dus dat een funksie op 'n keer omsluit word en die res met behulp van 0/1 veranderlikes slegs op hul bo- of ondergrense beskou word. Die beste oplossing vir elke funksie as 'n (tussen onder- en bogren) tob-funksie word op hierdie manier gekry. Die beste van al hierdie oplossings sal dan die globale optimale oplossing bevat. Hierdie benadering word in Algoritme 2.2 gegee.

Algoritme 2.2 (Tob-funksie-omsluiting)

Bereken die optimale toewysing van hulpbronne aan 'n aantal konvekse funksies oor wisselende intervalle.

1. [Inisieer]
Laat $t = 0$. Bereken vir elke funksie i die maksimum wat daardie funksie kan aanneem, dus $f_i(u_i)$. Laat die beste doelfunksie $z_b = 0$.
2. [Stel die tob-funksie vas]
Vermeerder t met 1. Die tob-funksie word aanvanklik met een lyn benader, dus

$k = 1$. Laat die ondergrens van die lyn $b_0 = l_t = 0$. Die bogrens van die lyn is $b_k = u_t$.

3. [Skep die LP]

(a) [Stel die doelfunksie van die heeltallige programmeringsprobleem saam]
Vir alle i , uitgesluit $i = t$ laat

$$\text{maks } z = \sum_{i=1, i \neq t}^n f_i(u_i)y_i + \sum_{j=1}^k \frac{f_t(b_j) - f_t(b_{j-1})}{b_j - b_{j-1}}(x_{tj} - b_{j-1})w_j + \sum_{j=1}^k f_t(b_{j-1})w_j.$$

(b) [Stel die beperkings vir die heeltallige programmeringsprobleem (HP) saam]

$$\sum_{i=1, i \neq t}^n u_i y_i + \sum_{j=1}^k w_j x_{tj} = B$$

$$\sum_{j=1}^k w_j = 1$$

$$x_{tj} \geq b_{j-1} \quad j = 1, 2, \dots, k$$

$$x_{tj} \leq b_j \quad j = 1, 2, \dots, k$$

$$y_i = 0/1 \text{ veranderlike} \quad i = 1, 2, 3, \dots, n \text{ en } i \neq t$$

$$w_j = 0/1 \text{ veranderlike} \quad j = 1, 2, \dots, k.$$

4. [Los die HP op en pas die HP aan, indien nodig]

In die oplossing moet net een van die w_j 's gelyk aan 1 wees. Gestel dit gebeur by $j = a$:

(a) [Indien $b_{a-1} < x_{ta} < b_a$] skep nog 'n lyn, dit wil sê $k = k + 1$ en vir $m = k$ tot $a + 1$ (inkrementeer negatief): $b_k = b_{k-1}$. Laat $b_a = x_{ta}$. Gaan na Stap 3.

(b) [Indien $x_{ta} = b_{a-1}$ of $x_{ta} = b_a$] gaan na Stap 5.

5. [Stoor die beste oplossing]

As $z_b < z$ dan $z_b = z$.

6. [Toets of klaar] As $t < n$ gaan na Stap 2, anders stop.

2.6.2.2 Algemene lineêre omsluiting

Die tweede benadering is om al die funksies gelyktydig te omsluit. Al die funksies word aanvanklik met 'n reguitlyn benader. Die funksie wat 'n oplossing tussen sy bo- en ondergrens gee, word dan vervang met een lyn meer totdat die omsluiting se oplossing saamval met die oorspronklike funksies. Dan is die optimum bereik. Hierdie benadering word in Algoritme 2.3 gegee.

Algoritme 2.3 (Algemene omsluiting)

Bereken die optimale toewysing van hulpbronne aan 'n aantal konvekse funksies oor wisselende intervalle.

1. [Inisieer]

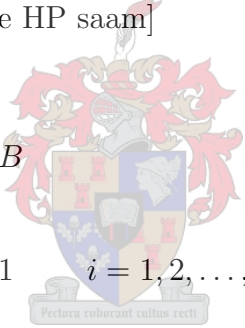
Laat die beste doelfunksie $z_b = 0$. Elke funksie $f_i(x_i)$ word aanvanklik met net een lyn benader, dus $k_i = 1$ vir $i = 1, 2, \dots, n$. Stel die grense van die intervalle waaroor die lyne strek, dus $b_{i0} = l_i = 0$ en $b_{i1} = u_i$.

2. [Skep die HP]

(a) [Stel die doelfunksie van die HP saam]

$$\text{maks } z = \sum_{i=1}^n \sum_{j=1}^{k_i} \frac{f_i(b_{ij}) - f_i(b_{ij-1})}{b_{ij} - b_{ij-1}} y_{ij} (x_{ij} - b_{ij-1}) + \sum_{i=1}^n \sum_{j=1}^{k_i} f_i(b_{ij-1}) y_{ij}$$

(b) [Stel die beperkings vir die HP saam]

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^{k_i} y_{ij} x_{ij} &= B \\ \sum_{j=1}^{k_i} y_{ij} &= 1 \quad i = 1, 2, \dots, n \\ x_{ij} &\geq b_{ij-1} \quad i = 1, 2, \dots, n \text{ en } j = 1, 2, \dots, k_i \\ x_{ij} &\leq b_{ij} \quad i = 1, 2, \dots, n \text{ en } j = 1, 2, \dots, k_i \\ y_{ij} &= 0/1 \quad i = 1, 2, 3, \dots, n \text{ en } j = 1, 2, \dots, k_i. \end{aligned}$$


3. [Los die HP op en pas die oplossing aan, indien nodig]

Vir $i = 1$ tot n :

(a) [Pas die HP aan, indien nodig]

In die oplossing moet daar vir elke i net een van die y_{ij} 's gelyk aan 1 wees. Gestel dit gebeur by $j = a$.

i. [As $b_{ia-1} < x_{ia} < b_{ia}$] skep nog 'n lyn vir funksie i , dit wil sê $k_i = k_i + 1$ en vir $m = k_i$ tot $a + 1$ (inkrementeer negatief): $b_{ik} = b_{ik-1}$. Laat $b_{ia} = x_{ia}$. Gaan na Stap 2.

ii. [Indien $x_{ia} = b_{ia-1}$ of $x_{ia} = b_{ia}$] vir $i = 1, 2, \dots, n$, gaan na Stap 4.

4. [Stoor die beste oplossing]

Indien $z_b < z$ dan $z_b = z$. Stop.

2.7 Nelineêre optimering

Beskou die mees algemene formulering van 'n nelineêre optimeringsprobleem

$$\text{maks } f(\underline{x}) \tag{2.9}$$

onderhewig aan

$$\underline{x} \in X$$

waar $f : X \rightarrow \mathbb{R}$ en $X \subseteq E^n$.

Die volgende definisies [34] is nodig om die teorie van nelineêre optimering te gebruik. 'n Delta-omgewing (δ -omgewing) van 'n punt \underline{x}^* word aangedui deur $N(\underline{x}^*, \delta)$.

Definisie 2.1 (Relatiewe maksimumpunt)

'n Relatiewe (lokale, plaaslike) maksimumpunt vir (2.9) is 'n punt $\underline{x}^* \in X$ waarvoor daar 'n δ bestaan sodat

$$f(\underline{x}) \leq f(\underline{x}^*) \quad \forall \underline{x} \in N(\underline{x}^*, \delta) \cap X.$$

Definisie 2.2 (Sterk relatiewe maksimumpunt)

'n Sterk relatiewe maksimumpunt vir (2.9) is 'n punt $\underline{x}^* \in X$ waarvoor daar 'n δ bestaan sodat

$$f(\underline{x}) < f(\underline{x}^*) \quad \forall \underline{x} \in N(\underline{x}^*, \delta) \cap X, \underline{x} \neq \underline{x}^*.$$

Definisie 2.3 (Globale maksimumpunt)

'n Globale maksimumpunt vir (2.9) is 'n punt $\underline{x}^* \in X$ waarvoor

$$f(\underline{x}) \leq f(\underline{x}^*) \quad \forall \underline{x} \in X.$$

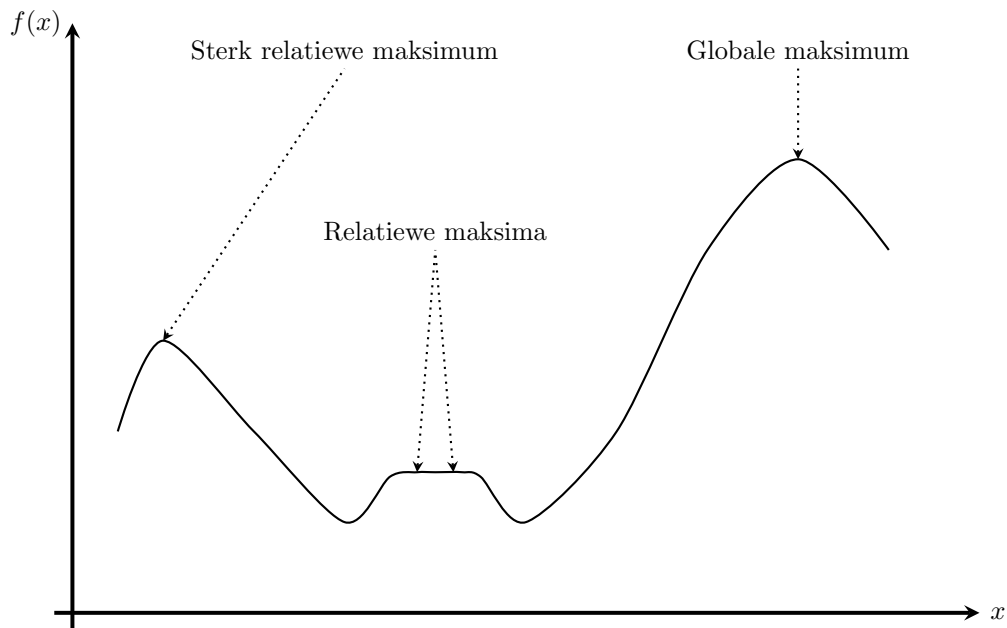
Indien die streng groter as hierbo geld, word \underline{x}^* 'n sterk globale maksimumpunt genoem.

Die skets in Figuur 2.2 gee 'n grafiese voorstelling van die begrippe wat in Definisies 2.1 tot 2.3 gedefinieer word.

2.7.1 Globale optimering

Om die algemene teorie vir globale optimering af te lei, het ons eers 'n aantal definisies nodig. Beskou die probleem

$$\text{maksimeer } f(\underline{x})$$



Figuur 2.2: Voorbeelde van tipes ekstreme punte.

onderhewig aan

$$\begin{aligned}
 h_j(\underline{x}) &= 0, & j &= 1, 2, \dots, m \\
 g_k(\underline{x}) &\geq 0, & k &= 1, 2, \dots, p \\
 \underline{x} &\in A \subseteq E^n,
 \end{aligned}
 \tag{2.10}$$

waar $m \leq n$ en die funksies f , h_j en $g_k \in C^2$ met $j = 1, 2, \dots, m$ en $k = 1, 2, \dots, p$.

Definisie 2.4 (Aktiewe en onaktiewe beperkings)

'n Beperking $g_k(\underline{x})$ is aktief by 'n toelaatbare punt \underline{x} as $g_k(\underline{x}) = 0$ en onaktief as $g_k(\underline{x}) > 0$.

Gelykheidsbeperkings is natuurlik altyd aktief.

Definisie 2.5 (Reguliere punt)

Gestel 'n punt \underline{x}^* bevredig die beperkings in (2.10). Definieer die volgende versameling indekse: $K = \{k \mid g_k(\underline{x}^*) = 0\}$. Dan is \underline{x}^* 'n reguliere punt indien die gradiëntvektore $\nabla h_j(\underline{x}^*)$ met $j = 1, 2, \dots, m$ en $\nabla g_k(\underline{x}^*)$ met $k \in K$ lineêr onafhanklik is.

'n Punt \underline{x}^* is dus regulier indien die gradiënte van die aktiewe beperkings by \underline{x}^* lineêr onafhanklik is. Met die definisie van 'n reguliere punt kan gewys word hoe 'n raakvlak op die aktiewe beperkings sal lyk.

Stelling 2.1 Gestel \underline{x}^* is 'n reguliere punt op die oppervlak S wat deur die gelykheidsbeperkings gedefinieer word. Dan is die raakvlak by \underline{x}^* gelyk aan

$$M = \{\underline{y} \mid \nabla h_j(\underline{x}^*)\underline{y} = 0 \quad \text{vir alle } j\}.$$

Die bewys vir Stelling 2.1 kan in [21] en [34] gevind word.

Met hierdie definisies en stellings kan die eerste orde nodige voorwaardes (Kuhn-Tucker) geformuleer word.

Stelling 2.2 (Kuhn-Tucker-voorwaardes)

Gestel \underline{x}^* is 'n relatiewe maksimumpunt vir (2.10) en 'n reguliere punt. Dan bestaan daar vektore $\underline{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ en $\underline{\mu} = (\mu_1, \mu_2, \dots, \mu_p)$ met $\mu_k \geq 0$ vir alle $k = 1, 2, \dots, p$, sodat

$$\frac{\partial f(\underline{x}^*)}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial h_j(\underline{x}^*)}{\partial x_i} + \sum_{k=1}^p \mu_k \frac{\partial g_k(\underline{x}^*)}{\partial x_i} = 0 \quad i = 1, 2, \dots, n \quad (2.11)$$

$$\mu_k g_k(\underline{x}^*) = 0 \quad k = 1, 2, \dots, p. \quad (2.12)$$

Die bewys vir Stelling 2.2 indien dit geformuleer is vir 'n minimeringsprobleem kan in [21] en [34] gevind word. Die ekwivalensie van Stelling 2.2 aan die stellings wat op bladsy 233 van Luenberger [21] en bladsy 2-27 van Sinclair [34] gegee word, word in Bylae A getoon.

In hierdie proefskrif, tensy spesifiek anders aangedui, dui die hoofletternotasie die Hessian van 'n funksie aan. Byvoorbeeld, gestel 'n funksie word gegee as $f(\underline{x})$, dan sal

$$F(\underline{x}) = \nabla^2 f(\underline{x}) = \begin{bmatrix} \frac{\partial^2 f(\underline{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_1 \partial x_j} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\underline{x})}{\partial x_i \partial x_1} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_i \partial x_j} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_i \partial x_n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\underline{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_n \partial x_j} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_n \partial x_n} \end{bmatrix}.$$

Die tweede orde nodige voorwaardes word in Stelling 2.3 gegee.

Stelling 2.3 (Tweede orde nodige voorwaardes)

Gestel f, h_1, h_2, \dots, h_m en $g_1, g_2, \dots, g_p \in C^2$ en \underline{x}^* is 'n relatiewe maksimumpunt vir (2.10) en 'n reguliere punt. Dan bestaan daar 'n $\underline{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ en $\underline{\mu} = (\mu_1, \mu_2, \dots, \mu_p)$ met $\mu_k \geq 0$ vir alle $k = 1, 2, \dots, p$, sodat (2.11) en (2.12) geld en sodat

$$L(\underline{x}^*) = F(\underline{x}^*) + \sum_{j=1}^m \lambda_j H_j(\underline{x}^*) + \sum_{k=1}^p \mu_k G_k(\underline{x}^*) \quad (2.13)$$

negatief semidefiniet is op die raakvlak van die aktiewe beperkings by \underline{x}^* .

Die bewys van Stelling 2.3, geformuleer vir 'n minimeringsprobleem, kan in [21] en [34] gevind word.

Die tweede orde voldoende voorwaardes word in Stelling 2.4 gegee.

Stelling 2.4 (Tweede orde voldoende voorwaardes)

Gestel f, h_1, h_2, \dots, h_m en $g_1, g_2, \dots, g_p \in C^2$. Gestel daar bestaan vektore $\underline{x}^* \in E^n$, $\underline{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ en $\underline{\mu} = (\mu_1, \mu_2, \dots, \mu_p)$ met $\mu_k \geq 0$ vir alle $k = 1, 2, \dots, p$ sodat

$$\frac{\partial f(\underline{x}^*)}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial h_j(\underline{x}^*)}{\partial x_i} + \sum_{k=1}^p \mu_k \frac{\partial g_k(\underline{x}^*)}{\partial x_i} = 0 \quad i = 1, 2, \dots, n \quad (2.14)$$

$$\mu_k g_k(\underline{x}^*) = 0 \quad k = 1, 2, \dots, p \quad (2.15)$$

$$\mu_k \geq 0 \quad k = 1, 2, \dots, p \quad (2.16)$$

en

$$L(\underline{x}^*) = F(\underline{x}^*) + \sum_{j=1}^m \lambda_j H_j(\underline{x}^*) + \sum_{k=1}^p \mu_k G_k(\underline{x}^*) \quad (2.17)$$

negatief definiet is op die raakvlak

$$M = \{ \underline{y} \mid \nabla h_j(\underline{x}^*) \underline{y} = 0, \nabla g_k(\underline{x}^*) \underline{y} = 0 \text{ vir alle } j = 1, 2, \dots, m \text{ en } k \in K \}$$

waar $K = \{k \mid g_k(\underline{x}^*) = 0, \mu_k > 0\}$. Dan is \underline{x}^* 'n sterk relatiewe maksimumpunt vir (2.10).

Die bewys vir Stelling 2.4, geformuleer vir 'n minimeringsprobleem, kan in [21] en [34] gevind word.

Indien die volgende stelling geld, is die voorwaardes wat in Stelling 2.2 gestel word ook die voldoende voorwaardes.

Stelling 2.5

Beskou die nielineêre programmeringsprobleem (NLPP) in (2.10). Indien $f(\underline{x})$ konkaaf is, en $h_j(\underline{x})$ vir alle $j = 1, 2, \dots, m$ en $g_k(\underline{x})$ vir alle $k = 1, 2, \dots, p$ is sodanig dat die toelaatbare gebied 'n konvekse versameling vorm, sal 'n punt \underline{x}^* wat die vergelykings in Stelling 2.2 bevredig, 'n optimale oplossing vir die NLPP in (2.10) wees.

Die bewys van hierdie stelling, geformuleer vir 'n minimeringsprobleem, kan in [34] gevind word.

2.8 Kuhn-Tucker-voorwaardes vir HTP

Die algemene formulering vir die probleem wat in hierdie studie beskou word, word in (1.1) gegee. Enige oplossing $\underline{x}^* = (x_1^*, x_2^*, \dots, x_i^*, \dots, x_n^*)$ van hierdie probleem kan met

behulp van drie versamelings, naamlik

$$\mathcal{L} = \{i|x_i^* = l_i\}, \quad (2.18)$$

$$\mathcal{U} = \{j|x_j^* = u_j\} \text{ en} \quad (2.19)$$

$$\mathcal{K} = \{k|x_k^* \in (l_k, u_k)\} \quad (2.20)$$

voorgestel word. Elkeen van die elemente van enige oplossing kan dus in net een van hierdie versamelings ingedeel word.

Die Kuhn-Tucker-voorwaardes vir 'n $N/A/G$ -HTP word in verskeie bronne gegee, sien byvoorbeeld [6, 7, 40]. Die Kuhn-Tucker-voorwaardes vir (1.1) is soortgelyk en word gegee deur (sien Bylae A)

$$f'_i(x_i) + \lambda + v_i - w_i = 0 \quad i = 1, 2, \dots, n \quad (2.21)$$

$$\nu_i(x_i - l_i) = 0 \quad i = 1, 2, \dots, n \quad (2.22)$$

$$\mu_i(-x_i + u_i) = 0 \quad i = 1, 2, \dots, n \quad (2.23)$$

$$\lambda \quad \text{teken onbeperk} \quad (2.24)$$

$$\nu_i \geq 0 \quad i = 1, 2, \dots, n \quad (2.25)$$

$$\mu_i \geq 0 \quad i = 1, 2, \dots, n. \quad (2.26)$$

Hierdie Kuhn-Tucker-voorwaardes is die nodige voorwaardes vir 'n oplossing om optimaal te wees.

Venter [40] toon vir 'n $N/A/G$ -HTP aan dat indien die versamelings (2.18) tot (2.20) gebruik word die Kuhn-Tucker-voorwaardes vereenvoudig kan word. Hierdie vereenvoudiging vir 'n $K/E/G$ -HTP word gegee deur

$$\begin{aligned} f'_i(x_i) + \lambda + \nu_i &= 0 \quad \forall i \in \mathcal{L} \text{ en } \nu_i \geq 0 \\ f'_j(x_j) + \lambda - \mu_j &= 0 \quad \forall j \in \mathcal{U} \text{ en } \mu_j \geq 0 \\ f'_k(x_k) + \lambda &= 0 \quad \forall k \in \mathcal{K}. \end{aligned}$$

Hieruit kan afgelei word dat, indien \underline{x}^* 'n optimale oplossing vir (1.1) is, sal geld dat [16, 40]

$$\begin{aligned} f'_i(x_i^*) &\leq -\lambda \quad \forall i \in \mathcal{L} \\ f'_j(x_j^*) &\geq -\lambda \quad \forall j \in \mathcal{U} \\ f'_k(x_k^*) &= -\lambda \quad \forall k \in \mathcal{K}. \end{aligned}$$

Venter [40] toon ook aan dat, indien die doelfunksie geminimeer moet word en die doelfunksie uit die som van skeibare konkawe funksies bestaan, \mathcal{K} hoogstens een element sal bevat. Wolvaardt [45] het ook 'n baie elegante bewys vir hierdie resultaat wat nie op die differensieerbaarheid van die funksies $f_i(x_i)$ berus nie.

2.9 Kromming van 'n funksie

Die begrip van die kromming van 'n kromme (of funksie) is goed gedokumenteer. Een so 'n voorbeeld is die definisie soos gegee deur Swokovski [35].

Definisie 2.6

Laat 'n gladde kromme C 'n parametrisering van $x = f(s)$, $y = g(s)$ hê. Laat θ die hoek tussen die raaklyn van C en die eenheidsvektor (\hat{i}) parallel aan die x -as wees. Die kromming van 'n kurwe word dan gedefinieer as

$$K = \left| \frac{d\theta}{ds} \right|.$$

Laat die kromme C gegee word deur 'n funksie f . Ons weet dat

$$\tan \theta = f' \quad \text{of} \quad \theta = \tan^{-1} f'. \quad (2.27)$$

Die funksie vir die booglengte $s(x)$ word gegee deur

$$s(x) = \int_a^x \sqrt{1 + (f')^2} dx, \quad (2.28)$$

waar a die x -koördinaat van 'n vaste punt op C is. Indien f'' bestaan, dan volg vanuit die kettingreël dat

$$\frac{d\theta}{dx} = \frac{d\theta}{ds} \frac{ds}{dx}$$

of

$$\frac{d\theta}{ds} = \frac{d\theta/dx}{ds/dx}$$

sodat vanuit Definisie 2.6 volg dat

$$K = \left| \frac{d\theta}{ds} \right| = \left| \frac{d\theta/dx}{ds/dx} \right|. \quad (2.29)$$

Uit (2.27) volg dat

$$\frac{d\theta}{ds} = \tan^{-1} f' = \frac{f''}{1 + (f')^2}$$

en uit (2.28) volg dat

$$\frac{ds}{dx} = \frac{d}{dx} \int_a^x \sqrt{1 + (f')^2} dx = \sqrt{1 + (f')^2}.$$

Indien bostaande in (2.29) gestel word, volg definisie 2.7.

Definisie 2.7

Indien C 'n gladde kromme van die grafiek $y = f(x)$ is, word die kromming (K) van C by die punt (x, y) gegee deur

$$K = \frac{|f''|}{[1 + (f')^2]^{3/2}}.$$

2.10 Primale simpleksalgoritme met bogrense

Die primale simpleksalgoritme met bogrense is 'n bekende algoritme wat in die meeste goeie operasionele navorsingsboeke verskyn (sien byvoorbeeld [13, 37, 44]). Die algoritme wat hier gegee word, is vereenvoudig om nog vinniger te wees indien met 'n LP wat net een beperking het, gewerk word. Van die redes vir hierdie vereenvoudiging is dat die oplossing van 'n lineêre HTP altyd sal bestaan, nooit onbegrens kan wees nie, ontarding nie kan plaasvind nie en daar slegs een simpleksiterasie nodig is om die optimale oplossing te bepaal.

Beskou probleem P, 'n lineêre HTP, hieronder.

$$\text{Maks } c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

onderhewig aan

$$\begin{aligned} x_1 + x_2 + \cdots + x_n &\leq b \\ 0 \leq x_i &\leq m_i \quad i = 1, 2, \dots, n. \end{aligned}$$

Uit die aard van die probleem wat in hierdie proefskrif aangebied word, sal $c \geq 0$ en $b \geq 0$.

'n Verduideliking van die begrip 'komplement van 'n veranderlike' is ook nodig voordat die algoritme gegee kan word. Indien 'n veranderlike, sê x_i , gekomplementeer word, beteken dit dat x_i met 'n veranderlike \tilde{x}_i vervang word, waar $\tilde{x}_i = m_i - x_i$ en $0 \leq \tilde{x}_i \leq m_i$. Indien die komplement van 'n veranderlike gekomplementeer word, word dit vervang met die oorspronklike veranderlike, oftewel $x_i = m_i - \tilde{x}_i$. Indien 'n veranderlike, sê x_t gekomplementeer word, verander die LP na

$$\text{maks } c_1x_1 + c_2x_2 + \cdots + c_t(m_t - \tilde{x}_t) + \cdots + c_nx_n$$

onderhewig aan

$$\begin{aligned} x_1 + x_2 + \cdots - \tilde{x}_t + \cdots + x_n &\leq b - m_t \\ 0 \leq x_i &\leq m_i \quad \text{vir } i = 1, 2, \dots, n; i \neq t \\ 0 \leq \tilde{x}_t &\leq m_t. \end{aligned}$$

Beskou die aanvanklike simplekstablo hieronder (vir probleem P) wat opgestel is vir die primale simpleksalgoritme vir bogrense:

	m_1	m_2	\dots	m_t	\dots	m_n	∞	
BV	c_1	c_2	\dots	c_t	\dots	c_n	0	RK
\downarrow	x_1	x_2	\dots	x_t	\dots	x_n	s	\downarrow
s	1	1	\dots	1	\dots	1	1	b
\underline{z}	$-c_1$	$-c_2$	\dots	$-c_t$	\dots	$-c_n$	0	

Dit is vir hierdie vereenvoudiging nie nodig om 'n g -ry te bereken nie omdat die z -ry altyd gelyk aan $-c$ gaan wees. Dit volg omdat

$$\begin{aligned} \underline{z} &= \underline{c}_{BV} B^{-1} A - \underline{c} \\ &= -\underline{c} \end{aligned}$$

Indien veranderlike x_t gekomplementeer word, het dit die volgende effek op die simpleks-tablo:

	m_1	m_2	\dots	m_t	\dots	m_n	∞	
BV	c_1	c_2	\dots	$-c_t$	\dots	c_n	0	RK
\downarrow	x_1	x_2	\dots	\tilde{x}_t	\dots	x_n	s	\downarrow
s	1	1	\dots	-1	\dots	1	1	$b - m_t$
\underline{z}	$-c_1$	$-c_2$	\dots	c_t	\dots	$-c_n$	0	

Dit wil sê die komplementering kan in die simplekstablo saamgevat word deur die volgende substitusies (die aksent dui die waarde van 'n veranderlike na komplementering aan):

1. Nuwe regterkant $b' = b - m_t$
2. $c'_t = -c_t$
3. $z'_t = -z_t$

Die vereenvoudigde primale simpleksalgoritme met bogrense om 'n lineêre HTP op los, word in Algoritme 2.4 gegee.

Algoritme 2.4 (HTP primale simpleksalgoritme met bogrense)

Bepaal die optimale oplossing vir 'n lineêre HTP.

1. [Toets vir optimaliteit]
Indien $z_i \geq 0$ vir alle $i = 1, 2, \dots, n$; is die tablo optimaal.
2. [Bepaal niebasisveranderlike om te komplementeer]
Kies $z_t = \min_i [z_i]$ vir alle $i = 1, 2, \dots, n$ gegewe dat $z_i < 0$. (Dit is dieselfde as om $c_t = \max_i [c_i]$ vir alle i sodat $z_i < 0$ te bepaal.)
3. [Komplementeer 'n veranderlike of doen 'n simpleksiterasie]
Indien $m_t \leq b$, gaan na Stap 3a, anders gaan na Stap 3b.
 - (a) [Komplementeer]
Komplementeer die veranderlike x_t , dit wil sê: $b' = b - m_t$, $c'_t = -c_t$ en $z'_t = -z_t$. Gaan na Stap 1.
 - (b) [Doen simpleksiterasie]
Doen 'n gewone simpleksiterasie met inkomende BV x_t en s wat die basis verlaat. Stop, die tablo is optimaal.

2.11 Duale simpleksalgoritme met bogrense

Anders as die primale simpleksalgoritme met bogrense is die duale simpleksalgoritme met bogrense heelwat swakker gedokumenteer. Wagner [43] het die eerste artikel wat hierdie algoritme beskryf, gepubliseer. De Kock [10] het ook 'n aangepaste weergawe van hierdie algoritme gebruik in die oplossing van lineêre programmeringsprobleme om veevoermengsels te optimeer. Visagie [42] gebruik 'n effens gewysigde vorm van De Kock se algoritme in die bepaling van optimale wisselboustelsels. (Sien [42] vir 'n bespreking van presies hoe Visagie en Wagner se algoritmes van mekaar verskil.) Die algoritme wat hier gegee word, is 'n vereenvoudiging van Visagie [42] se algoritme. Die vereenvoudiging volg uit die feit dat 'n LP wat altyd begrens is, geen siklusse bevat nie en slegs een beperking het, opgelos moet word.

Die vereenvoudigde duale simpleksalgoritme met bogrense om 'n lineêre HTP op los, word in Algoritme 2.5 gegee. Die vereenvoudigings volg soortgelyk aan dié van die primale algoritme.

Algoritme 2.5 (HTP duale simpleksalgoritme met bogrense)

Bepaal die optimale oplossing vir 'n lineêre HTP.

1. [Komplementeer al die veranderlikes x_i vir alle $i = 1, 2, \dots, n$.] Komplementeer die veranderlike x_i deur die volgende substitusies uit te voer: $b' = b - m_i$, $c'_i = -c_i$ en $z'_i = -z_i$.
2. [Toets vir optimaliteit]
Indien $b \geq 0$ is die tablo optimaal.
3. [Bepaal niebasisveranderlike om te komplementeer]
Kies $z_t = \min_i [z_i]$ vir alle $i = 1, 2, \dots, n$ gegewe dat $z_i \geq 0$. (Dit is dieselfde as om $c_t = \min_i [c_i]$ vir alle i sodat $z_i \geq 0$ te bepaal.)
4. [Komplementeer 'n veranderlike of doen 'n simpleksiterasie]
Indien $m_t \leq |-b|$, gaan na Stap 4a, anders gaan na Stap 4b.
 - (a) [Komplementeer]
Komplementeer die veranderlike \tilde{x}_t , dit wil sê: $b' = b + m_t$, $c'_t = -c_t$ en $z'_t = -z_t$. Gaan na Stap 2.
 - (b) [Doen simpleksiterasie]
Doen 'n gewone simpleksiterasie met inkomende BV \tilde{x}_t en s wat die basis verlaat. Stop, die tablo is optimaal.

2.12 Samevatting

In hierdie hoofstuk is twee hoofpunte aangespreek. Die eerste hoofpunt handel oor reeds gepubliseerde werk wat spesifiek is tot HTP's wat in hierdie proefskrif aangespreek word.

Dit verskaf dus 'n bondige oorsig van wat reeds in die literatuur rakende hierdie HTP's bestaan. Die tweede hoofpunt wat aangespreek word, handel oor 'n paar reeds gepubliseerde kernidees rakende nielineêre programmering, spesiale gevalle van die simpleks-algoritme en kalkulus. Hierdie kernidees is nodig in die ontwikkeling van stellings en algoritmes in die hoofstukke wat volg.



Hoofstuk 3

$K/E/V$ -HTP's

In hierdie hoofstuk word die probleem soos in (1.1) beskou, met die ekstra voorwaardes dat al die skeibare komponente van die doelfunksie konveks en monotoon stygend is, en al die intervalle waaroor die veranderlikes x_i kan varieer dieselfde is. Dit beteken $K/E/V$ -HTP's word beskou. Die meer algemene probleem waar die veranderlikes wisselende intervalle het, word eers beskou, daarna word die spesiale geval van $K/E/V$ -HTP's beskou.

3.1 Kuhn-Tucker-voorwaardes vir $K/E/G$

Die Kuhn-Tucker-voorwaardes wat in (2.21) tot (2.26) gegee word, vereenvoudig vir $K/E/G$ -HTP's tot

$$f'_l(x_l) \leq -\lambda \quad \forall l \in \mathcal{L} \quad (3.1)$$

$$f'_u(x_u) \geq -\lambda \quad \forall u \in \mathcal{U} \quad (3.2)$$

$$f'_k(x_k) = -\lambda \quad \forall k \in \mathcal{K} \quad (3.3)$$

soos deur Venter [40] en Ibaraki & Katoh [16] aangetoon vir die minimering van konkawe funksies. Venter [40] toon ook vir $N/A/G$ -HTP's aan dat 'n optimale oplossing hoogstens een element in die versameling \mathcal{K} sal hê. Die soortgelyke resultaat vir $K/E/G$ -HTP's volg in Stelling 3.1.

Stelling 3.1 (Nodige voorwaardes vir $K/E/G$)

Indien $\underline{x}^* = (x_1, x_2, \dots, x_i, \dots, x_n)$ 'n optimale oplossing van 'n $K/E/G$ -HTP is, sal \underline{x}^* (3.1) tot (3.3) bevredig en daar sal hoogstens 1 element van \underline{x}^* in versameling \mathcal{K} wees.

3.2 Vaste interval

Beskou eers die eenvoudiger probleem waar die volgende voorwaardes op (1.1) geplaas word:

- Al die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$ in die doelfunksie is konveks en stygend;
- die ondergrense van al die veranderlikes is nul, dit is $l_i = 0$ en $f_i(0) = 0$ met $i = 1, 2, \dots, n$; en
- die bogrense van al die veranderlikes is dieselfde, dit is $u_i = u$ met $i = 1, 2, \dots, n$.

Die tweede voorwaarde hierbo lyk op sigwaarde baie beperkend, maar indien in ag geneem word dat al die funksies getransformeer kan word sodat hul ondergrens gelyk aan nul is, is dit in wese glad nie die geval nie.

Hierdie voorwaardes gee aanleiding tot 'n $K/E/V$ -HTP. Hierdie probleem is hoofsaaklik om twee redes eenvoudiger as die algemene formulering:

- al die funksies in die doelfunksie het dieselfde vorm; en
- volgens Stelling 3.1 sal net een van die funksies 'n waarde aanneem wat nie gelyk aan die onder- of bogrens van die ooreenkomstige veranderlike is nie. Hierdie waarde, $R \in (l, u) = (0, u)$, kan vooraf bepaal word.

Uit Stelling 3.1 volg dat elke funksie in die doelfunksie by die optimale antwoord óf sy waarde by sy ondergrens óf sy waarde by die bogrens óf sy waarde tussen die onder- en bogrens (hoogstens een funksie) kan aanneem. Die sterk resultaat van Stelling 3.1 is dus dat dit die $K/E/G$ -HTP lineier. Omdat daar vir elke funksie $f_i(x_i)$ met $i = 1, 2, \dots, n$ net drie waardes van belang is, naamlik $f_i(l)$, $f_i(R)$ en $f_i(u)$, kan die funksie $f_i(x_i)$ vervang word met 'n lyn wat deur die punte $(l, f_i(l))$ en $(R, f_i(R))$ gaan asook 'n lyn deur $(R, f_i(R))$ en $(u, f_i(u))$. Die liniëring van die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$ word in Figuur 3.1 getoon. Indien versameling \mathcal{K} die leë versameling is, sal $R = 0$, wat die probleem dan nog verder vereenvoudig deurdat alle funksies dan slegs deur een lyn wat $(l, f_i(u))$ en $(u, f_i(u))$ verbind, omsluit word. In hierdie geval word die funksies slegs by hulle bogrense vasgemaak vanaf die funksie met die lyn met die grootste helling totdat die funksie met die laaste helling gebruik word, of die hulpbron uitgeput is.

Die algemene vergelyking vir lyn₁ is

$$f_i^1(x_i) = \frac{f_i(R) - f_i(l)}{R - l} x_i$$

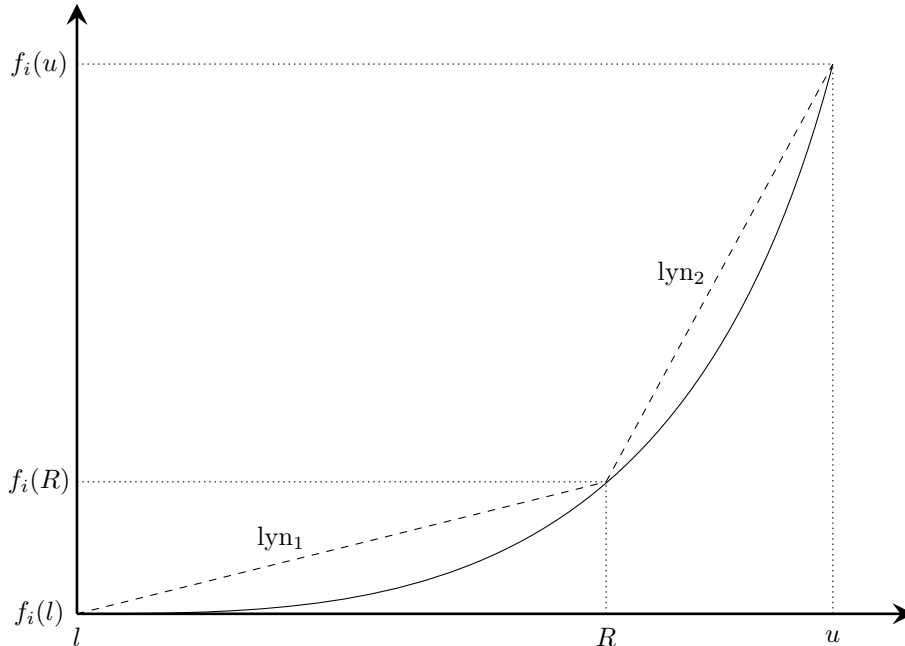
wat vereenvoudig na

$$f_i^1(x_i) = \frac{f_i(R)}{R} x_i$$

omdat aanvaar word dat al die funksies $f_i(x_i)$ in die oorsprong begin. Die vergelyking vir lyn₂ is

$$f_i^2(x_i) = \frac{f_i(u) - f_i(R)}{u - R} (x_i - R) + f_i(R) - f_i(u)$$

Indien die regterkant van die beperking so 'n waarde aanneem dat $R = 0$ is, kan die resulterende enkele lyn uit enige een van hierdie twee vergelykings gekry word. Indien die



Figuur 3.1: Die liniëring van $f_i(x_i)$.

limiet met $R \rightarrow u$ vir die vergelyking van lyn_1 geneem word, of die limiet met $R \rightarrow l$ vir die vergelyking van lyn_2 geneem word, word die resulterende enkele lyn in die algemeen gegee deur

$$f_i^0(x_i) = \frac{f_i(u) - f_i(l)}{u - l} x_i.$$

wat vereenvoudig tot

$$f_i^0(x_i) = \frac{f_i(u)}{u} x_i$$

omdat aanvaar word dat al die funksies $f_i(x_i)$ in die oorsprong begin. Die volgende definisies is nodig. Die definisies word in die algemeen gegee, maar sal natuurlik vereenvoudig vir 'n $K/E/V$ -HTP.

Definisie 3.1 (Onderhelling)

Die onderhelling by 'n punt R , met $R \in (l_i, u_i)$, word gegee deur

$${}^o h_i(R) = \frac{f_i(R) - f_i(l_i)}{R_i - l_i}.$$

Dit is die helling van die lyn wat die punte $(l_i, f_i(l_i))$ en $(R_i, f_i(R_i))$ verbind.

Definisie 3.2 (Bohelling)

Die bohelling by 'n punt R , met $R \in (l_i, u_i)$, word gegee deur

$${}^b h_i(R) = \frac{f_i(u_i) - f_i(R_i)}{u_i - R_i}.$$

Dit is die helling van die lyn wat die punte $(R_i, f_i(R_i))$ en $(u_i, f_i(u_i))$ verbind.

Definisie 3.3 (Gemiddelde helling)

Die gemiddelde helling,

$$gh_i = \frac{f_i(u_i) - f_i(l_i)}{u_i - l_i},$$

is die helling van die lyn wat die punte $(l_i, f_i(l_i))$ en $(u_i, f_i(u_i))$ verbind.

Volgens Stelling 3.1 is dit 'n nodige voorwaarde vir optimaliteit van 'n $K/E/G$ -HTP dat daar hoogstens een funksie, sê funksie k , kan wees waarvan dit waar is dat daardie funksie nie 'n toedeling van die hulpbron kry wat nie gelyk aan die onder- of bogrens van daardie funksie is nie. Hierdie resultaat gee aanleiding tot die volgende definisie.

Definisie 3.4 (tob-funksie)

Indien funksie r die funksie is wat 'n toedeling uit die hulpbron ontvang wat tussen die onder- en bogrens van x_r is, word hierdie funksie die tob-funksie (tussen onder- en bogrens) genoem.

3.3 Heeltallige programmering as oplossingsmetode vir $K/E/V$

Die feit dat daar vir die $K/E/V$ -HTP slegs drie punte van belang is, impliseer dat hierdie probleem eintlik 'n kombinatoriese optimeringsprobleem is waarin:

- net een van die drie punte $(l, f_i(l)) = (0, 0)$, $(R, f_i(R))$ en $(u, f_i(u))$ vir elke funksie i gekies moet word [Elke funksie moet 'n waarde óf by sy ondergrens óf by sy bogrens óf by R aanneem];
- net een punt $(R, f_i(R))$ uit al die funksies i gekies moet word [Volgens Stelling 3.1 kan daar hoogstens een so 'n funksie wees];
- die hoeveelheid toegedeel gelyk aan die hulpbron moet wees $[R + \sum_{i \in \mathcal{U}} u = B]$; en
- die doelfunksie 'n maksimum aanneem [maksimeer: $\sum_{i \in \mathcal{L}} f_i(0) + f_r(R) + \sum_{i \in \mathcal{U}} f_i(u)$].

Hierdie kombinatoriese probleem kan met behulp van heeltallige programmering (HP) opgelos word. Hierdie formulering word gegee deur

$$\text{maks} \sum_{i=1}^n w_i f_i(R) + \sum_{i=1}^n y_i f_i(u)$$

onderhewig aan

$$\begin{aligned}
 w_i + y_i &\leq 1 & i = 1, 2, \dots, n \\
 \sum_{i=1}^n w_i &= 1 \\
 R \sum_{i=1}^n w_i + u \sum_{i=1}^n y_i &= B \\
 w_i, y_i & & 0, 1 \text{ veranderlikes,}
 \end{aligned} \tag{3.4}$$

waar $w_i = 1$ indien funksie i die waarde by R aanneem en $w_i = 0$ andersins en $y_i = 1$ indien funksie i 'n toedeling by sy bogrens kry en $y_i = 0$ andersins.

Die formulering in (3.4) kan egter vereenvoudig word. Die liniëring laat die idee van die gebruik van lineêre programmering (LP) ontstaan. 'n Baie eenvoudige en elegante LP-formulering kan gemaak word indien opgelet word dat een van die twee lyne wat die funksie definieer altyd bindend is. Indien die toedeling aan 'n funksie by sy ondergrens of R is, is lyn₁ van daardie funksie bindend en indien die toedeling aan 'n funksie by sy bogrens of by R is, is lyn₂ van daardie funksie bindend. Die gemengde heeltallige programmeringsprobleem kan dan geformuleer word as

onderhewig aan

$$\begin{aligned}
 &\text{maks } \sum_{i=1}^n V_i \\
 V_i - \frac{f_i(R) + f_i(0)}{R} x_i &\leq M y_i & i = 1, 2, \dots, n \\
 V_i - \frac{f_i(u) - f_i(R)}{u - R} x_i - f_i(R) + \frac{f_i(u) - f_i(R)}{u - R} R &\leq M(1 - y_i) & i = 1, 2, \dots, n \\
 \sum_{i=1}^n x_i &= B \\
 x_i &\leq u & i = 1, 2, \dots, n \\
 V_i, x_i &\geq 0 & i = 1, 2, \dots, n \\
 y_i & & 0, 1 \text{ veranderlike,}
 \end{aligned} \tag{3.5}$$

waar x_i die hoeveelheid toegedeel aan funksie i , V_i die opbrengs van funksie i , M 'n groot getal en y_i 'n binêre veranderlike is. Indien $y_i = 0$, word die beperking van lyn₁ van funksie i bindend en lyn₂ van funksie i word verslap. Indien $y_i = 1$, word die beperking van lyn₂ van funksie i bindend en lyn₁ van funksie i word verslap.

3.4 Breukalgoritme as oplossingsmetode vir $K/E/V$

Die breukalgoritme is 'n eenvoudige algoritme wat met brute krag al die funksies een na die ander as 'n element van die versameling \mathcal{K} beskou en dan die res van die funksies se

toedelings óf op hul bogrens óf op hul ondergrens vasmaak. Voor die algoritme geformuleer kan word, het ons eers 'n verdere resultaat nodig wat dit makliker maak om die funksies wat nie 'n element van \mathcal{K} is nie in te deel in \mathcal{U} of \mathcal{L} . Hierdie resultaat word in Stelling 3.2 gegee. Die probleem met vaste intervale kom nie algemeen in die literatuur voor nie en hierdie resultaat is nuut.

'n Nodige voorwaarde vir optimaliteit van 'n oplossing van die $K/E/V$ -HTP is dat die gemiddelde helling van elke funksie wat 'n element van \mathcal{U} is, groter of gelyk aan die gemiddelde hellings van elke funksie wat 'n element van \mathcal{L} is, moet wees.

Stelling 3.2 (Uitruilstelling)

Gestel $\underline{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$ is 'n oplossing van $K/E/V$. 'n Nodige voorwaarde vir \underline{x}^* om optimaal te wees, is dat

$$\min_{\forall r \in \mathcal{U}} [g_{h_r}] \geq \max_{\forall s \in \mathcal{L}} [g_{h_s}]. \quad (3.6)$$

Bewys: (Teenspraak) Gestel \underline{x}^* is wel optimaal maar (3.6) geld nie. Dit beteken dat daar 'n $\alpha \in \mathcal{U}$ en 'n $\beta \in \mathcal{L}$ moet wees sodat

$$\begin{aligned} g_{h_\alpha} &< g_{h_\beta} \\ \frac{f_\alpha(u_\alpha) - f_\alpha(l_\alpha)}{u_\alpha - l_\alpha} &< \frac{f_\beta(u_\beta) - f_\beta(l_\beta)}{u_\beta - l_\beta} \\ f_\alpha(u_\alpha) - f_\alpha(l_\alpha) &< f_\beta(u_\beta) - f_\beta(l_\beta) \\ f_\alpha(u_\alpha) + f_\beta(l_\beta) &< f_\beta(u_\beta) + f_\alpha(l_\alpha). \end{aligned}$$

Die doelfunksiewaarde van \underline{x}^* word gegee deur

$$D^* = f_\alpha(u_\alpha) + f_\beta(l_\beta) + R,$$

waar R die som van die res van die terme in die doelfunksie is.

Skep nou 'n nuwe oplossing \underline{x}' wat net van \underline{x}^* verskil in dié opsig dat α en β uitgeruil word. Dit wil sê $\alpha \in \mathcal{L}$ en 'n $\beta \in \mathcal{U}$, oftewel $x'_\alpha = l_\alpha$ en $x'_\beta = u_\beta$. Hierdie nuwe oplossing \underline{x}' se doelfunksiewaarde word gegee deur

$$\begin{aligned} D' &= f_\beta(u_\beta) + f_\alpha(l_\alpha) + R \\ &> f_\alpha(u_\alpha) + f_\beta(l_\beta) + R \\ &= D^*. \end{aligned}$$

Die feit dat $D' > D^*$ lewer die verlangde teenspraak omdat \underline{x}^* die optimale oplossing is. ■

Hier volg 'n beskrywing van die stappe van die breukalgoritme vir vaste intervale:

Bereken die optimale toewysing van hulpbronne aan 'n aantal konvekse funksies oor vaste intervale as volg:

1. Bereken die gemiddelde helling vir al die funksies en rangskik hierdie funksies in 'n versameling A van die funksie met die grootste gemiddelde helling tot die funksie met die kleinste gemiddelde helling. Laat $f_1(x_1)$ die funksie met die grootste gemiddelde helling, $f_2(x_2)$ die funksie met die tweede grootste gemiddelde helling, ens. wees. $A = \{f_1(x_1), f_2(x_2), \dots, f_n(x_n)\}$. Bereken ook die toedeling aan die toefunksie, naamlik R . Laat $i = 1$.
2. Deel R eenhede van die hulpbron aan funksie i toe. Plaas funksie i oor in die versameling \mathcal{K} . Stel funksies in volgorde van grootste na kleinste gemiddelde helling gelyk aan hul bogrens en plaas hulle oor na die versameling \mathcal{U} totdat die hulpbron uitgeput is. Plaas die oorblywende funksies in A oor na \mathcal{L} .
3. Indien $i = n$, gaan na Stap 4. So nie, stel $i = i + 1$ en herhaal Stap 2.
4. Bereken die doelfunksiewaarde vir elkeen van die n oplossings wat in Stap 2 bereken is en kies die oplossing met die grootste doelfunksiewaarde. Hierdie oplossing is die optimale oplossing omdat alle oplossings wat aan die nodige voorwaardes voldoen, in ag geneem is. Stop.

Vervolgens word die pseudokode vir die breukalgoritme in Algoritme 3.1 gegee. Die notasie wat in Algoritmes 3.1 en 3.2 gebruik word, word aan die begin van die proefskrif gegee.

Algoritme 3.1 (Breukalgoritme vir vaste intervale)

Prosedure BREUKALGORITME(HTP)

$D_t \leftarrow 0$

$D_i \leftarrow 0$

$D_m \leftarrow 0$

vir $i \leftarrow 1$ **tot** n **doen** \triangleright Sorteër die funksies van grootste tot kleinste gemiddelde helling

vir $j \leftarrow 1$ **tot** $(n - i)$ **doen**

as $(f_j(u) - f_j(l))/(u - l) < (f_{j+1}(u) - f_{j+1}(l))/(u - l)$ **dan**

$k \leftarrow I_g(j)$

$I_g(j) \leftarrow I_g(j + 1)$

$I_g(j + 1) \leftarrow k$

end as

end vir

end vir

vir $i \leftarrow 1$ **tot** n **doen**

\triangleright Bepaal die waarde van die breuk en A_b

as $(i)(u) = B$ **dan**

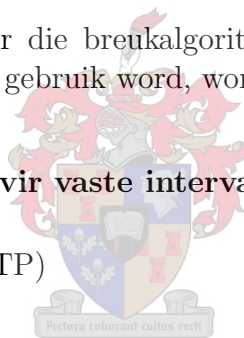
$R \leftarrow 0$

$A_b \leftarrow i$

ontsnap vir

anders as $(i)(u) > B$ **dan**

$R \leftarrow B - (i - 1)(u)$

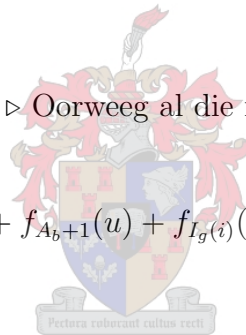


```

     $A_b \leftarrow i - 1$ 
    ontsnap vir
  end as
end vir
as  $A_b u > B$  dan
  Stop: "Die hulpbron is te veel"
end as
as  $R = 0$  dan
  vir  $i \leftarrow 1$  tot  $A_b$  doen
     $x_i^m = u$ 
  end vir
  Stop: "Optimale oplossing is gevind"
end as
vir  $i \leftarrow 1$  tot  $n$  doen
  as  $i < A_b$  dan
     $D_t \leftarrow D_t + f_{I_g(i)}(u)$ 
  anders
     $D_t \leftarrow D_t + f_{I_g(i)}(l)$ 
  end as
end vir
vir  $i \leftarrow 1$  tot  $n$  doen
  as  $i \leq A_b$  dan
     $D_{I_g(i)} \leftarrow D_t - f_{I_g(i)}(u) + f_{A_b+1}(u) + f_{I_g(i)}(R)$ 
  anders
     $D_i = D_t + f_{I_g(i)}(R)$ 
  end as
  as  $D_m < D_i$  dan
     $D_m \leftarrow D_{I_g(i)}$ 
     $k \leftarrow i$ 
  end as
end vir
vir  $i \leftarrow 1$  tot  $k - 1$  doen
   $x_{I_g(i)}^m \leftarrow u$ 
end vir
vir  $i \leftarrow k + 1$  tot  $n$  doen
   $x_{I_g(i)}^m \leftarrow l$ 
end vir
 $x_{I_g(k)}^m \leftarrow R$ 
Stop: "Optimale oplossing is gevind"
end Procedure

```

▷ Toets of die hulpbron te veel is om toe te deel
 ▷ Eerste A_b funksies kry 'n toedeling gelyk aan u
 ▷ Kry 'n beginoplossing
 ▷ Oorweeg al die funksies ($f_i(x_i)$) een vir een as die tob-funksie



3.5 Hellingalgoritme as oplossingsmetode vir $K/E/V$

Voor die hellingalgoritme afgelei kan word, moet 'n verdere nodige voorwaarde vir 'n oplossing om optimaal te wees, eers afgelei word. 'n Nodige voorwaarde vir optimaliteit van 'n oplossing van 'n $K/E/V$ -HTP is dat die bohelling by die punt R van die funksie wat in \mathcal{K} is, kleiner of gelyk moet wees aan die bohelling by R van alle funksies wat 'n element van \mathcal{U} is; en dat die onderhelling van die funksie in \mathcal{K} groter of gelyk moet wees aan die onderhelling by R van al die funksies wat in \mathcal{L} is.

Stelling 3.3 (Nodige voorwaarde vir $K/E/V$)

Gestel $\underline{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$ is 'n oplossing van $K/E/V$, met funksie k wat 'n toedeling van x_k^* kry, waar $x_k^* \in (l_k, u_k)$. 'n Nodige voorwaarde vir \underline{x}^* om optimaal te wees, is dat

$${}^b h_k(x_k^*) \leq \min_{\forall r \in \mathcal{U}} [{}^b h_r(x_k^*)], \quad (3.7)$$

en

$${}^o h_k(x_k^*) \geq \max_{\forall s \in \mathcal{L}} [{}^o h_s(x_k^*)]. \quad (3.8)$$

Bewys: (Teenspraak) Beskou eers vergelyking (3.7). Gestel \underline{x}^* is wel optimaal, met $x_k^* \in (l_i, u_i)$, maar (3.7) geld nie. Dit beteken daar moet iewers 'n $\alpha \in \mathcal{U}$ wees sodat

$$\begin{aligned} {}^b h_\alpha(x_k^*) &< {}^b h_k(x_k^*) \\ \frac{f_\alpha(u_\alpha) - f_\alpha(x_k^*)}{u_\alpha - x_k^*} &< \frac{f_k(u_k) - f_k(x_k^*)}{u_k - x_k^*} \\ f_\alpha(u_\alpha) - f_\alpha(x_k^*) &< f_k(u_k) - f_k(x_k^*) \\ f_\alpha(u_\alpha) + f_k(x_k^*) &< f_k(u_k) + f_\alpha(x_k^*) \end{aligned}$$

Die doelfunksiewaarde vir die oplossing \underline{x}^* is

$$D^* = f_\alpha(u_\alpha) + f_k(x_k^*) + R,$$

waar R die som van die res van die terme in die doelfunksie is.

Skep 'n nuwe oplossing \underline{x}' wat van \underline{x}^* verskil net in dié opsig dat die indekse α en k uitgeruil word. Dit wil sê $\alpha \in \mathcal{K}$ en $k \in \mathcal{U}$. Die doelfunksiewaarde van hierdie nuwe oplossing word gegee deur

$$\begin{aligned} D' &= f_k(u_k) + f_\alpha(x_k^*) + R \\ &> f_\alpha(u_\alpha) + f_k(x_k^*) + R \\ &= D^* \end{aligned}$$

Die feit dat $D' > D^*$ lewer die verlangde teenspraak omdat \underline{x}^* die optimale oplossing is.

Beskou nou vergelyking (3.8). Gestel \underline{x}^* is wel optimaal, maar (3.8) geld nie. Dit beteken daar moet iewers 'n $\beta \in \mathcal{L}$ wees sodat

$$\begin{aligned} \mathcal{H}_\beta(x_k^*) &> \mathcal{H}_k(x_k^*) \\ \frac{f_\beta(x_k^*) - f_\beta(l_\beta)}{x_k^* - l_\beta} &> \frac{f_k(x_k^*) - f_k(l_k)}{x_k^* - l_k} \\ f_\beta(x_k^*) - f_\beta(l_\beta) &> f_k(x_k^*) - f_k(l_k) \\ f_\beta(x_k^*) + f_k(l_k) &> f_k(x_k^*) + f_\beta(l_\beta) \end{aligned}$$

Die doelfunksiewaarde vir die oplossing \underline{x}^* is

$$D^* = f_k(x_k^*) + f_\beta(l_\beta) + R,$$

waar R die som van die res van die terme in die doelfunksie is.

Skep 'n nuwe oplossing \underline{x}'' wat van \underline{x}^* verskil net in dié opsig dat die indekse β en k uitgeruil word. Dit wil sê $\beta \in \mathcal{K}$ en $k \in \mathcal{L}$. Die doelfunksiewaarde van hierdie nuwe oplossing word gegee deur

$$\begin{aligned} D'' &= f_\beta(x_k^*) + f_k(l_k) + R \\ &> f_k(x_k^*) + f_\beta(l_\beta) + R \\ &= D^* \end{aligned}$$

Die feit dat $D'' > D^*$ lewer die verlangde teenspraak omdat \underline{x}^* die optimale oplossing is. ■

Hier volg 'n beskrywing van die stappe vir 'n algoritme wat die nodige voorwaardes in Stelling 3.3 gebruik om die optimale toedeling van 'n hulpbron aan n aktiwiteite met monotoon stygende, konvekse opbrengsfunksies te bepaal:

Bereken die optimale toewysing van hulpbronne aan 'n aantal konvekse funksies oor vaste intervalle:

1. Bereken die grootte van die breuk R , asook die aantal funksies wat op hulle bogrense vasgestel moet word. Gestel daar is A_b sulke funksies.
 - (a) Bereken die gemiddelde helling (\mathcal{H}_i) vir al die funksies en rangskik dit in 'n versameling \mathcal{G} van die grootste tot die kleinste.
 - (b) Bereken die onderhelling ($\mathcal{H}_i(R)$) vir al die funksies en rangskik dit in 'n versameling \mathcal{P} van die grootste tot die kleinste.
 - (c) Bereken die bohelling ($\mathcal{H}_i(R)$) vir al die funksies en rangskik dit in 'n versameling \mathcal{C} van die kleinste tot die grootste.
2. Laat \mathcal{P}' die subversameling van \mathcal{P} wees wat die eerste $A_b + 1$ gerangskikte elemente van \mathcal{P} bevat. Laat \mathcal{C}' die subversameling van \mathcal{C} wees wat die eerste $n - A_b$ gerangskikte elemente van \mathcal{C} bevat. Skep nou die versameling \mathcal{K}' wat al die elemente (funksies) bevat wat in beide \mathcal{P}' en \mathcal{C}' is. Slegs die funksies wat in \mathcal{K}' is, is kandidate vir die funksie wat die toedeling van R eenhede van die hulpbron kry. Laat die elemente van $\mathcal{K}' = \{f_1(x_1), \dots, f_v(x_v), \dots, f_{k'}(x_{k'})\}$ wees. Stel $v = 1$.

3. Laat $x_v = R$ en laat $\mathcal{G}' = \mathcal{G} - f_v(x_v)$. Stel die eerste A_b funksies in \mathcal{G}' se toedelings gelyk aan hul bogrense. Al die ander funksies se toedelings word op hul ondergrense gestel.
4. Indien $v = k'$, gaan na Stap 5, anders laat $v = v + 1$ en herhaal Stap 3.
5. Kies die oplossing met die grootste doelfunksiewaarde uit die k' oplossings wat in Stap 3 bereken is. Hierdie oplossing is die optimale oplossing, omdat alle oplossings wat aan die nodige voorwaardes voldoen, ondersoek is.

Vervolgens word die pseudokode van hierdie algoritme in Algoritme 3.2 gegee.

Algoritme 3.2 (Hellingalgoritme vir vaste intervalle)

Procedure HELLINGALGORITME(HTP)

$D_t \leftarrow 0$

$D_i \leftarrow 0$

$D_m \leftarrow 0$

vir $I \leftarrow 1$ **tot** n **doen**

$g_{h_i} \leftarrow \frac{f_i(u) - f_i(l)}{u - l}$

$I_g(i) \leftarrow i$

$o_{h_i}(R) \leftarrow \frac{f_i(R) - f_i(l)}{R - l}$

$I_o(i) \leftarrow i$

$b_{h_i}(R) \leftarrow \frac{f_i(u) - f_i(R)}{u - R}$

$I_b(i) \leftarrow i$

end vir

vir $i \leftarrow 1$ **tot** n **doen** ▷ Sorteër die funksies van grootste tot kleinste helling

vir $j \leftarrow 1$ **tot** $(n - i)$ **doen**

as $g_{h_j} < g_{h_{j+1}}$ **dan**

$k \leftarrow I_g(j)$

▷ Ruil funksies j en $j + 1$ om

$I_g(j) \leftarrow I_g(j + 1)$

$I_g(j + 1) \leftarrow k$

end as

end vir

end vir

vir $i \leftarrow 1$ **tot** n **doen** ▷ Bepaal die waarde van die breuk en A_b

as $(i)(u) = B$ **dan**

$R \leftarrow 0$

$A_b \leftarrow i$

Ontsnap vir

anders as $(i)(u) > B$ **dan**

$R \leftarrow B - (i - 1)(u)$

$A_b \leftarrow i - 1$

Ontsnap vir

end as

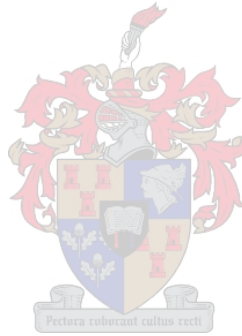
end vir



```

as  $A_b u > B$  dan ▷ Toets of die hulpbron te veel is om toe te deel
  Stop: "Die hulpbron is te veel"
end as
as  $R = 0$  dan ▷ Eerste  $A_b$  funksies kry 'n toedeling gelyk aan  $u$ 
  vir  $i \leftarrow 1$  tot  $A_b$  doen
     $x_{I_g(i)}^m = u$ 
  end vir
  Stop: "Optimale oplossing is gevind"
end as
vir  $i \leftarrow 1$  tot  $n$  doen ▷ Sorteert onder- en bohellings
  vir  $j \leftarrow 1$  tot  $n - i$  doen
    as  ${}^o h_j(R) < {}^o h_{j+1}(R)$  dan
       $k \leftarrow I_o(j)$ 
       $I_o(j) \leftarrow I_o(j + 1)$ 
       $I_o(j + 1) \leftarrow k$ 
    end as
    as  ${}^b h_j(R) > {}^b h_{j+1}(R)$  dan
       $k \leftarrow I_b(j)$ 
       $I_b(j) \leftarrow I_b(j + 1)$ 
       $I_b(j + 1) \leftarrow k$ 
    end as
  end vir
end vir
vir  $i \leftarrow 1$  tot  $A_b + 1$  doen
   $O_h(I_o(i)) \leftarrow 1$ 
end vir
vir  $i \leftarrow 1$  tot  $n - A_b$  doen
   $B_h(I_b(i)) \leftarrow 1$ 
end vir
 $j \leftarrow 0$ 
vir  $i \leftarrow 1$  tot  $n$  doen
  as  $O_h(i) = 1$  and  $B_h(i) = 1$  dan
     $j \leftarrow j + 1$ 
     $Kand(j) = i$ 
  end as
end vir
 $D_t = 0$ 
vir  $i \leftarrow 1$  tot  $A_b$  doen
   $D_t \leftarrow D_t + f_{I_g(i)}(u)$ 
end vir
vir  $i \leftarrow 1$  tot  $j$  doen
  as  $Kand(i) < A_b$  dan
     $D \leftarrow D_t + f_{I_g(A_b+1)-f_{Kand(i)}(u)+f_{Kand(i)}(R)}$ 
  anders
     $D \leftarrow D_t + f_{Kand(i)}(R)$ 

```



```

end as
as  $D > D_m$  dan
   $D_m \leftarrow D$ 
   $r \leftarrow Kand(i)$ 
end as
end vir
end Prosedure

```

3.6 Samevatting

Hierdie afdeling bestaan uit drie onderafdelings. Die eerste onderafdeling beskryf die metode hoe die resultate verkry is, die tweede onderafdeling verskaf die resultate en in die laaste onderafdeling word 'n aantal kort gevolgtrekkings oor die resultate gemaak.

3.6.1 Metodologie

In sommige gevalle bestaan daar 'n stel standaard numeriese probleme waarteen nuwe algoritmes getoets kan word om hulle prestasie teen bestaande algoritmes te bepaal. Daar bestaan byvoorbeeld sulke stelle van numeriese voorbeelde vir die handelsreisigerprobleem, sekere versamelings van skeduleringsprobleme en sekere versamelings van houerpakprobleme (Eng.: *bin packing problems*). Daar bestaan nie so 'n standaard stel van HTP's waarteen die nuwe algoritmes wat in hierdie proefskrif voorgestel word, gemeet kan word nie. In die afwesigheid van so 'n stel probleme word probleme in hierdie proefskrif lukraak gegenereer en opgelos om 'n gemiddelde oplossingstyd vir hierdie stel lukraak gegenereerde probleme te bepaal.

Excel se *randbetween*-funksie is gebruik om lukraak 'n hulpbron tussen 0 en nu , 'n gemiddelde helling (wat saam met die waardes vir m_i en c_i herlei word na die regte waarde vir s_i), 'n funksienommer (om een van die vier families van konvekse funksies te kies) en waardes vir m_i en c_i (hul waardes stel die vorm van die funksie) te genereer.

Dit is belangrik om daarop te let dat weens 'n programmeringsbeperking¹ die oplossingstye vir die vertak-en-begrens-algoritme eerder as 'n ondergrens op die oplossingstye beskou moet word.

Die lukraak gegenereerde probleme is met behulp van 'n Pentium(R) 4-rekenaar met 'n 3.2 GHz sentrale verwerkingseenheid (Eng.: *CPU*) en 1 GB lees-en-skryf-geheue (Eng.:

¹Daar word 'n maksimum van 64 KB se geheue aan 'n subroetine in VBA toegeken. Dit het die implikasie dat daar in die vertak-en-begrens-algoritme hoogstens 32 000 subprobleme onthou kan word. Hierdie probleem kan die hoof gebied word deur die subprobleme oor meer as een subroetine te versprei, maar dit sal heelwat meer programmeringswerk impliseer wat niks nuwe kennis tot die studie bydra nie. As gevolg van hierdie beperking word die vertak-en-begrens-algoritme dus gestop nadat dit 32 000 subprobleme opgelos het. Die implikasie hiervan is dat die gewone vertak-en-begrens-algoritme dus in enkele gevalle langer kan neem as wat die getalle in Tabel 3-1 aandui. Hierdie waardes moet dus eerder as 'n ondergrens vir die oplossingstyd beskou word.

RAM) opgelos. Die rekenaarkode is in Excel met VBA geskep om die algoritmes te programmeer. Die kode vir hierdie algoritmes kan op die CD agterin hierdie proefskrif in die lêer met lêernaam `kev.xls` gevind word.

3.6.2 Resultate

Die resultate vir prestasie van die verskillende algoritmes vir die oplossing van $K/E/V$ -HTP word in Tabel 3-1 gegee. Dieselfde 100 funksies is vir elke keuse van n met al vier die algoritmes opgelos.

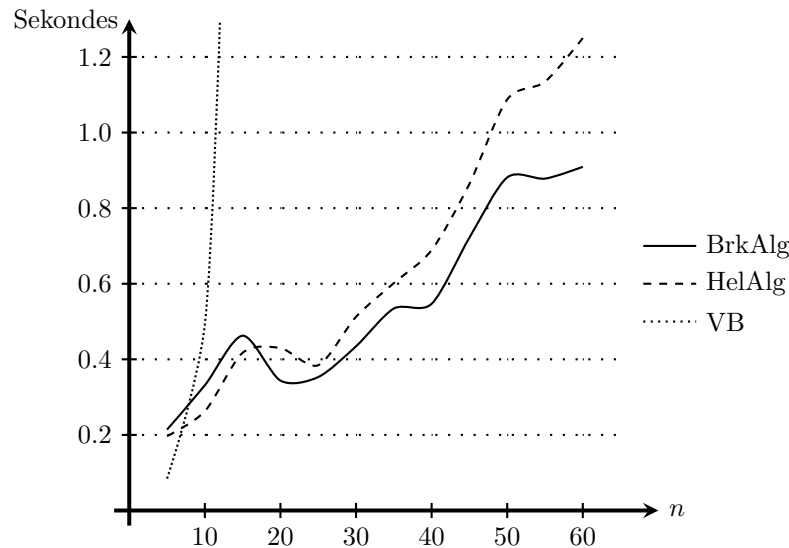
n	Din prog	BrkAlg	HelAlg	VB
5	14.404	0.268	0.246	0.105
10	95.711	0.414	0.328	0.615
15	266.314	0.578	0.521	5.371
20	535.115	0.430	0.537	17.955
25	–	0.441	0.480	70.270
30	–	0.543	0.641	111.605
35	–	0.668	0.752	150.873
40	–	0.684	0.861	199.834
45	–	0.902	1.080	266.549
50	–	1.102	1.359	284.754
55	–	1.098	1.418	350.547
60	–	1.137	1.563	397.117

Tabel 3-1: Totale oplossingstyd in sekondes van 100 lukraak gegenereerde $K/E/V$ -HTP's. Hierdie HTP's bestaan uit n funksies in die doelfunksie, wat lukraak gegenereer is uit die vier families van konvekse funksies. 'n Gemiddelde helling van tussen 5 en 8 word lukraak vir elke funksie gegenereer. 'n Vaste intervallengte van 100 eenhede is gebruik. Die volgende afkortings word gebruik: Din prog = dinamiese programmering, BrkAlg = breukalgoritme, HelAlg = hellingalgoritme en VB = vertak-en-begrens-algoritme.

Die gemiddelde tyd wat dit elk van die algoritmes neem om 'n $K/E/V$ -HTP met n funksies op te los, word in Figuur 3.2 gegee. Die grafiek vir dinamiese programmering word nie getoon nie, omdat die waardes daarvoor te groot is en nie op dieselfde skaal voorgestel kan word nie.

'n Ander manier om algoritmes met mekaar te vergelyk, is om te kyk na die persentasie van die tyd wat sekere algoritme(s) neem om dieselfde probleme as 'n bakenalgoritme op te los. Vir hierdie doel is die vertak-en-begrens-algoritme gekies as die bakenalgoritme en word die breuk-, sowel as die hellingalgoritme daarteen gemeet. Hierdie resultate volg in Tabel 3-2 en word grafies in Figuur 3.3 getoon.

Dit is uit Tabel 3-1 en Figuur 3.2 duidelik dat dinamiese programmering die swakste vaar om die $K/E/V$ -HTP op te los. Aangesien die oplossingstyd vir hierdie algoritme so vinnig toeneem as 'n funksie van n , is die resultate $n = 30$ tot $n = 60$ nie gegenereer nie, omdat hierdie algoritme te lank neem om die probleem op te los. Verder gebeur dit ook dat hierdie metode meer as die toegelate 64 KB geheue per subroetine gebruik.



Figuur 3.2: Grafieke van die gemiddelde tyd wat dit vier verskillende algoritmes neem om 'n $K/E/V$ -HTP met n funksies op te los. Die volgende afkortings word in die sleutel gebruik: *BrkAlg* = breukalgoritme, *HelAlg* = hellingalgoritme en *VB* = vertak-en-begrens-algoritme.

'n Punt waarop gelet moet word, is dat die oplossingstyd van die probleem met behulp van dinamiese programmering 'n funksie is van die intervallengte (dit is $u_i - l_i$). Hoe langer hierdie interval is, hoe langer neem dinamiese programmering om die probleem op te los. Vir die eksperimente in Tabel 3-1 is die vaste interval dus vasgestel op 100 eenhede. Al drie die ander algoritmes se oplossingstyd is nie 'n funksie van die intervallengte nie (sien Tabel 3-3). Dinamiese programmering se kompleksiteit is ook 'n funksie van B . Hoe nader B aan $\frac{1}{2} \sum_{i=1}^n u_i$ is, hoe langer neem dinamiese programmering om die optimale oplossing van $K/E/G$ -HTP te bepaal (sien Figuur 3.4).

3.6.3 Gevolgtrekkings

Uit die resultate in §3.6.2 is dit duidelik dat beide die breukalgoritme sowel as die hellingalgoritme beter resultate lewer as die bestaande oplossingsmetodes, naamlik dinamiese programmering en die vertak-en-begrens-metode. Dinamiese programmering is ordes stadiger as die ander metodes en is dus 'n swak keuse as maatstaf. Ander redes wat dinamiese programmering 'n swak keuse as maatstaf maak, is die feit dat dinamiese programmering se kompleksiteit 'n funksie van beide die intervallengte, naamlik $(u_i - l_i)$, en die grootte van die regterkant B van die hulpbronbeperking is. Omdat B een van die veranderlikes is wat lukraak gekies word, beïnvloed hierdie keuse die kompleksiteit van die dinamiese-programmeringsalgoritme. In die res van hierdie proefskrif sal dinamiese programmering dus nie gebruik word as maatstaf nie, behalwe waar ander maatstawwe nie moontlik is nie.

In werklikheid is die kompleksiteit van dinamiese programmering orde die aantal boë in die dinamiese-programmeringsnetwerk. As in ag geneem word dat die aantal boë in die

n	BrkAlg	HelAlg	VB
5	253.704	233.333	100
10	67.302	53.333	100
15	10.764	9.709	100
20	2.393	2.991	100
25	0.628	0.684	100
30	0.487	0.574	100
35	0.443	0.498	100
40	0.342	0.431	100
45	0.339	0.405	100
50	0.387	0.477	100
55	0.313	0.405	100
60	0.286	0.393	100

Tabel 3-2: Die oplossingstyd van die breukalgoritme (*BrkAlg*) en die hellingalgoritme (*HelAlg*) as 'n persentasie van die oplossingstyd van die vertak-en-begrens-algoritme (*VB*).

$u_i - l_i$	Din prog	BrkAlg	HelAlg	VB
10	2.175	1.992	2.238	1.625
50	8.097	2.265	2.101	1.738
100	21.593	2.171	2.210	1.578
150	44.578	2.230	2.140	1.582
200	87.812	2.289	2.171	1.695

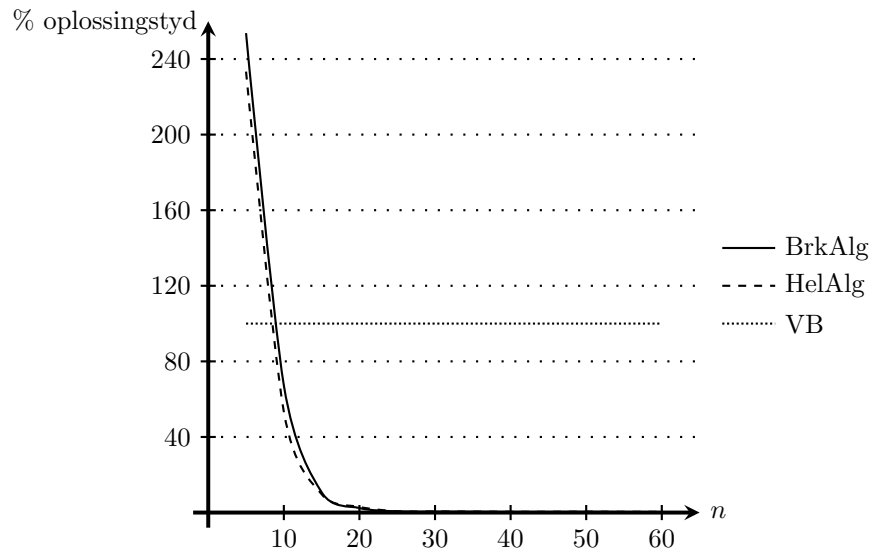
Tabel 3-3: Totale oplossingstyd in sekondes van 100 lukraak gegenereerde $K/E/V$ -HTP's met 10 funksies in die doelfunksie teenoor 'n toenemende intervallengte waaroor die veranderlikes strek. Die volgende afkortings word gebruik: *Din prog* = dinamiese programmering, *BrkAlg* = breukalgoritme, *HelAlg* = hellingalgoritme en *VB* = vertak-en-begrens-algoritme.

netwerk toeneem met 'n toename in beide n en B , verklaar dit hoekom die tydkompleksiteit van dinamiese programmering toeneem met 'n toename in beide n en B .

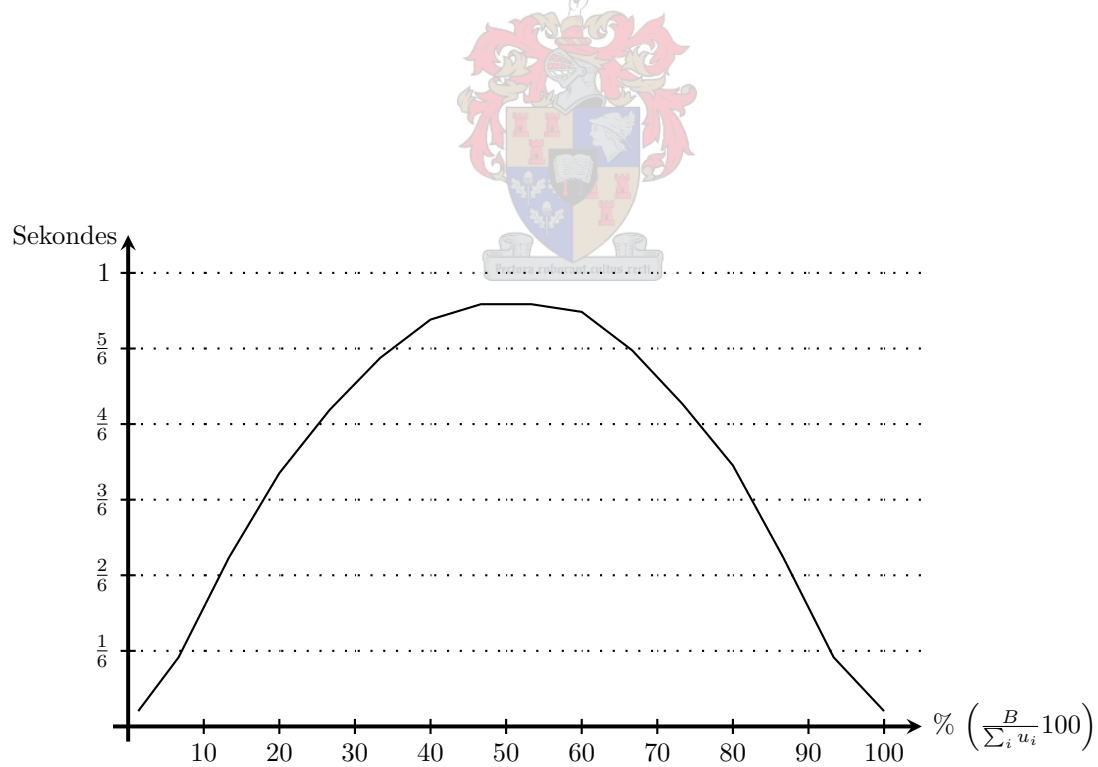
Dit is interessant om daarop te let dat, indien funksies nie uit vier families van funksies gegenereer word nie, maar net uit polinomiese funksies van die vorm

$$f_i(x_i) = a_i x_i^{b_i},$$

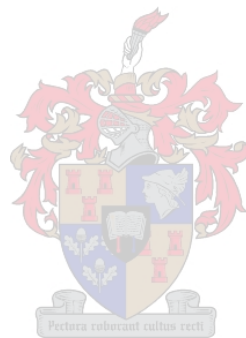
is die oplossingstye vir die breuk- en hellingalgoritme effens stadiger as die tye in Tabel 3-1, terwyl die oplossingstye vir die vertak-en-begrens-algoritme rofweg dieselfde bly.



Figuur 3.3: Die oplossingstye van die breukalgoritme (*BrkAlg*) en die hellingalgoritme (*HelAlg*) uitgedruk relatief tot die oplossingstye van die vertak-en-begrens-algoritme.



Figuur 3.4: Oplossingstyd van 'n *K/E/V-HTP* met behulp van dinamiese programmering vir verskillende waardes van *B* uitgedruk as 'n persentasie van die som van die bogrense. Vir hierdie data was $n = 15$ en $u_i - l_i = 100$.



Hoofstuk 4

$K/E/G$: Soekalgoritmes

Die struikelblok met algoritmes vir $K/E/G$ is dat daar nie vir hierdie probleem voldoende voorwaardes vir optimaliteit bepaal kan word nie. Die gevolg hiervan is dat klassieke kontinue nielineêre optimeringsmetodes nie van veel nut gaan wees vir 'n $K/E/G$ -HTP nie. Daarom moet daar na alternatiewe maniere gesoek word om die probleem te benader. In hierdie hoofstuk word soekalgoritmes voorgestel vir die oplossing van $K/E/G$ -HTP's.

4.1 Teoretiese beskouing

Die algemene formulering van die $K/E/G$ -HTP word gegee deur

$$\text{maks } f(\underline{x}) = \sum_{i=1}^n f_i(x_i)$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^n x_i &= B \\ 0 \leq x_i &\leq u_i \quad i = 1, 2, \dots, n, \end{aligned} \tag{4.1}$$

met alle funksies $f_i(x_i)$ konvekse funksies.

Die Kuhn-Tucker-voorwaardes vir hierdie formulering word in (3.1) tot (3.3) gegee. Hieronder volg die tweede orde nodige voorwaardes.

4.1.1 Tweede orde nodige voorwaardes

Beskou die probleem in (4.1). Vir hierdie formulering volg vanuit Stellings 2.3 en 2.4 dat, vir 'n oplossing (x_1^*, \dots, x_n^*) om optimaal te wees, daar 'n $\lambda \in \mathbb{R}$ en $v_1, \dots, v_n, w_1, \dots, w_n \geq$

0 moet bestaan, sodat

$$\sum_{i=1}^n x_i^* = B \quad (4.2)$$

$$f'(x_i^*) + \lambda + v_i - w_i = 0 \quad (4.3)$$

$$v_i(x_i^*) = 0 \quad i = 1, \dots, n \quad (4.4)$$

$$w_i(u_i - x_i^*) = 0 \quad i = 1, \dots, n. \quad (4.5)$$

Verder moet

$$L(\underline{x}^*) = F(\underline{x}^*) = \begin{bmatrix} F_1(x_1^*) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & F_n(x_n^*) \end{bmatrix}, \quad (4.6)$$

negatief semidefiniet op die raakvlak M vir die bindende beperkings wees. Die matriks $F(\underline{x}^*)$ se nienul-elemente is

$$F_i(x_i^*) = \frac{\partial^2 f_i(x_i^*)}{\partial x_i^2}.$$

Dit beteken dat vir alle waardes $z_1, z_2, \dots, z_n \in M$ moet

$$[z_1, \dots, z_n] \begin{bmatrix} F_1(x_1^*) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & F_n(x_n^*) \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \leq 0 \quad (4.7)$$

$$z_1^2 F_1(x_1^*) + \dots + z_n^2 F_n(x_n^*) \leq 0, \quad (4.8)$$

waar $F_i(x_i^*) = \frac{\partial^2 f_i(x_i^*)}{\partial x_i^2}$.

Let daarop dat $M = M_1 \cap M_2 \cap M_3$. Dit volg dus dat vir $z_i \in M$ moet

$$z_1 + z_2 + \dots + z_n = 0 \quad M_1 \quad (4.9)$$

$$z_i = 0 \text{ as } x_i^* = 0 \quad M_2 \quad (4.10)$$

$$z_i = 0 \text{ as } x_i^* = u_i \quad M_3. \quad (4.11)$$

Uit die tweede orde nodige voorwaardes kan ons affei dat daar vir die $K/E/G$ -HTP nie voldoende voorwaardes vir optimaliteit bepaal kan word nie. Hierdie resultaat word in Stelling 4.1 gegee.

Stelling 4.1

Daar kan nie voldoende voorwaardes bepaal word sodat 'n toelaatbare oplossing vir 'n $K/E/G$ -HTP optimaal is nie.

Bewys: Die bewys vir hierdie stelling volg direk uit (4.8) en (4.9). Die enigste oplossing vir

$$z_1^2 F_1(x_1^*) + \dots + z_n^2 F_n(x_n^*) \leq 0$$

en

$$z_1 + z_2 + \cdots + z_n = 0$$

is $z_1 = z_2 = \cdots = z_n = 0$ omdat alle $F_i(x_i) > 0$ vir $x_i \in (l_i, u_i)$. Die funksie $L(\underline{x}^*)$ kan dus nooit negatief definiet wees nie, wat beteken dat daar nie voldoende voorwaardes bepaal kan word nie. ■

4.1.2 'n Ander blik op Stelling 3.1

'n Meer elegante formulering en bewys vir Stelling 3.1 kan nou gegee word.

Stelling 4.2

'n Optimale oplossing van 'n $K/E/G$ -HTP kan hoogstens een element in \mathcal{K} hê; dit is hoogstens een tob-funksie hê.

Bewys: As $\underline{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$ 'n optimale oplossing 'n $K/E/G$ -HTP is, volg vanuit die tweede orde nodige voorwaardes dat (4.8) moet geld vir alle moontlike waardes van z_i soos gedefinieer deur (4.9) tot (4.11). Let op dat vir alle $i = 1, 2, \dots, n$ moet $F_i(x_i) > 0$ indien $x_i \in (l_i, u_i)$ omdat al die funksies $f_i(x_i)$ streng konveks is oor hierdie interval. Beskou die volgende drie gevalle wat alle moontlike oplossings van 'n $K/E/G$ -HTP dek.

1. Alle veranderlikes is gelyk aan hul onder- of bogrense.
Dit beteken $x_i^* = l_i$ of u_i vir alle $i = 1, 2, \dots, n$. Volgens (4.10) en (4.11) sal $z_i = 0$ vir alle $i = 1, 2, \dots, n$. In hierdie geval word vergelyking (4.8) bevredig vir alle moontlike waardes wat die z_i 's op die raakvlak kan aanneem.
Hierdie geval impliseer dat \mathcal{K} 'n leë versameling sal wees.
2. Een veranderlike is ongelyk aan sy onder- en bogrens.
Dit beteken dat $x_s \in (l_s, u_s)$ en $x_i^* = l_i$ of u_i vir alle $i = 1, 2, \dots, n, i \neq s$. In hierdie geval volg vir alle $i = 1, 2, \dots, n$ uitgesluit $i \neq s$ vanuit (4.10) en (4.11) dat $z_i = 0$. Verder moet $z_s = 0$ omdat (4.9) moet geld. In hierdie geval word vergelyking (4.8) ook bevredig vir alle moontlike waardes wat die z_i 's op die raakvlak kan aanneem.
In hierdie geval is daar dus een element in versameling \mathcal{K} .
3. Twee of meer veranderlikes in ongelyk aan hul onder- en bogrense.
Aanvaar sonder verlies van algemeenheid dat die eerste s veranderlikes ongelyk aan hul grense is. Dit beteken dat $x_k \in (l_k, u_k)$ met $k = 1, 2, \dots, s$ en $2 \leq s \leq n$. Verder sal $x_i^* = l_i$ of u_i vir alle $i = s + 1, s + 2, \dots, n$. Volgens (4.10) en (4.11) moet $z_i = 0$ vir $i = s + 1, s + 2, \dots, n$ en vanuit (4.9) volg dat $z_1 + z_2 + \cdots + z_s = 0$. Dit impliseer dat daar oneindig baie oplossings vir z_1 tot z_s op die raakvlak M bestaan. Al hierdie waardes van z_1 tot z_s bevredig egter nie vergelyking (4.8) nie¹.

Hierdie geval is dus nie moontlik nie.

¹Vergelyking (4.8) word wel bevredig in die enkele spesiale geval waar $z_i = 0$ vir alle $i = 1, 2, \dots, n$, maar (4.8) moet bevredig word vir alle waardes wat die z_i 's op die raakvlak kan aanneem.

Slegs punt 1 of punt 2 hierbo is moontlik en daarom kan daar hoogstens een element in \mathcal{K} wees. ■

4.2 Soekalgoritmes

Die filosofie waarop hierdie soekalgoritmes berus, is om elke funksie $f_i(x_i)$ om die beurt as die tob-funksie te beskou en die optimale oplossing met daardie funksie as tob-funksie te bepaal. Die eerste algoritme (breukalgoritme) wat hier aangebied word, is 'n veralgemening van die breukalgoritme vir vaste intervale. Hierdie algoritme is egter nie effektief nie en kom in wese neer op 'n brute-krag-benadering. Dit verdien nie werklik verdere aandag nie en die stappe daarvan word net kortliks hieronder gelys. Die hellingalgoritme vir vaste intervale kan egter nie uitgebrei word na die wisselende intervale nie. Die blaarsoek- en taksoekalgoritmes wat daarna aangebied word, is verfynings van die breukalgoritme vir wisselende intervale. Beide hierdie algoritmes poog om die soektog na die optimale toedeling, nadat die tob-funksie vasgestel is, te versnel.

4.2.1 Breukalgoritme [brute krag]

Die breukalgoritme vir vaste intervale kan uitgebrei word om wisselende intervale ook te akkommodeer. Dit verg heelwat meer berekenings om die optimale toedeling van die hulpbron te bepaal nadat die funksie in \mathcal{K} is vasgestel is. Twee komplikasies tree hier in. Nadat die res ($\notin \mathcal{K}$) van die funksies op hulle bo- en/of ondergrense ingedeel is, kan daar funksie(s) op hul bogrense wees wat eerder op hul ondergrense vasgepen moet word. Die volle hoeveelheid hulpbron van hierdie funksie(s) word dan by die toedeling van die funksie in \mathcal{K} bygevoeg. Daar kan ook funksie(s) wees waarvan die toedeling by hulle ondergrens is, wat eerder by hul bogrens moet wees. Hierdie hoeveelheid hulpbron moet dan by die funksie in \mathcal{K} se toedeling afgetrek word.

Algoritme 4.1 (Breukalgoritme vir wisselende intervale)

Bereken die optimale toewysing van hulpbronne aan 'n aantal konvekse funksies oor wisselende intervale.

1. [Stel alle funksies se toedelings gelyk aan hul ondergrense en verminder die hulpbron met hierdie hoeveelheid.]
Stel vir alle i : $x_i = l_i = 0$. Laat $i = 1$.
2. [Maak alle moontlike ruilings van funksies tussen versamelings I en J sodat funksie i die breuk bly en stoor die beste van hierdie oplossings.]

Genereer alle moontlike oplossings:

j	x_1	\dots	x_{i-1}	x_{i+1}	\dots	x_{n-2}	x_{n-1}	x_n	x_i
1	l_1	\dots	l_{i-1}	l_{i+1}	\dots	l_{n-2}	l_{n-1}	l_n	$B - l_1 - \dots - l_{i-1} - l_{i+1} - \dots - l_{n-1} - l_n$
2	l_1	\dots	l_{i-1}	l_{i+1}	\dots	l_{n-2}	l_{n-1}	u_n	$B - l_1 - \dots - l_{i-1} - l_{i+1} - \dots - l_{n-1} - u_n$
3	l_1	\dots	l_{i-1}	l_{i+1}	\dots	l_{n-2}	u_{n-1}	l_n	$B - l_1 - \dots - l_{i-1} - l_{i+1} - \dots - u_{n-1} - l_n$
4	l_1	\dots	l_{i-1}	l_{i+1}	\dots	l_{n-2}	u_{n-1}	u_n	$B - l_1 - \dots - l_{i-1} - l_{i+1} - \dots - u_{n-1} - u_n$
\vdots	\vdots	\dots	\vdots	\vdots	\dots	\vdots	\vdots	\vdots	\vdots
2^{n-1}	u_1	\dots	u_{i-1}	u_{i+1}	\dots	u_{n-2}	u_{n-1}	u_n	$B - u_1 - \dots - u_{i-1} - u_{i+1} - \dots - u_{n-1} - u_n$

Kies die grootste doelfunksiewaarde uit die doelfunksiewaardes wat met die oplossings hierbo ooreenstem waarvoor die waarde van x_i tussen die bo- en ondergrens is. Dit beteken kies

$$O_i \leftarrow \max_j \{f_1(x_1) + \dots + f(x_{i-1}) + f(x_{i+1}) + \dots + f(x_n)\}.$$

- [Draai die indeks i aan, of stop.]
As $i < n$: Laat $i = i + 1$ en gaan na Stap 2. Indien $i = n$: Stop en gaan na Stap 4.
- [Kies die beste oplossing uit die oplossings wat gedurende elke iterasie in Stap 2 gekies is.]
Dit wil sê kies

$$\text{Optimale oplossing} \leftarrow \max_i \{O_i\}.$$

'n Nadeel van Algoritme 4.1 is dat dit in wese met brute krag al die moontlike oplossings genereer wat aan die nodige voorwaardes in Stelling 3.1 voldoen. Die kompleksiteit van hierdie soektog neem vinnig toe namate die aantal funksies vermeerder. Spesifiek het hierdie soektog 'n kompleksiteit van $O(n2^{n-1})$, waar n die aantal funksies is. Die soektog in Algoritme 4.1 kan as 'n binêre boom beskou word. In hierdie binêre boom stem elke vertakking (in 0 of 1) ooreen met 'n keuse van of die betrokke aktiwiteit op sy onder- of bogrens vasgestel (onderskeidelik 0 of 1) gaan word of nie. Al die takke in hierdie boom stem nie noodwendig ooreen met toelaatbare oplossings nie. Die gevolg hiervan is dat van hierdie takke verwyder kan word sodat die boom (en soektog) dus minder kompleks is. In die volgende twee algoritmes word gepoog om hierdie soektog kleiner (vinniger) te maak.

4.2.2 Blaarsoekalgoritme

Elke funksie word om die beurt gekies om die funksie te wees wat 'n toedeling tussen sy bo- en ondergrense kry. Hierdie funksie (tob-funksie) word uit die probleem gehaal en die binêre boom word met die orige $n - 1$ funksies opgestel. Die blaarsoekalgoritme stem ooreen met 'n diepte eerste benadering deurdat die hele tak tot by die blaar beskou word. Die algoritme soek dus op 'n gestruktureerde wyse in die blare vir die beste oplossing. Die hele tak van die binêre boom, van die oorsprong tot by die blaar, beskryf

'n moontlike oplossing. Die tak word dus gebruik om te bepaal of daardie tak se oplossing 'n toelaatbare oplossing lewer. Elke oplossing wat met so 'n tak (van oorsprong tot blaar) ooreen stem, lewer een van drie tipes oplossings:

1. Toelaatbare oplossing
Dit beteken die hulpbron word presies opgebruik deur die $n - 1$ funksies in die boom, of daar bly net genoeg oor sodat die res toegedeel kan word aan die funksie wat 'n toedeling tussen sy bo- en ondergrens kry.
2. Ontoelaatbare oplossing (met te min van die hulpbron)
In hierdie geval benodig die toedeling aan die $n - 1$ funksies in die boom meer van die hulpbron as wat beskikbaar is.
3. Ontoelaatbare oplossing (met te veel van die hulpbron)
In hierdie geval bly daar meer as die bogrens (van die funksie wat 'n toedeling tussen sy onder- en bogrens moet kry) van die hulpbron oor nadat die hulpbron aan elkeen van die $n - 1$ funksies in die boom toegeken is.

Indien 'n toelaatbare oplossing verkry word, word die doelfunksie bereken, anders word die oplossing aangepas na 'n hoër blaar toe. Hoeveel hoër beweeg word, word bepaal deur in watter van die bostaande drie situasies die oplossing val. Op dié manier word elke tak dus sistematies ondersoek. Indien alle moontlike toelaatbare oplossings bereik is, word die algoritme gestop.

Die datastruktuur om oplossings voor te stel is 'n 0/1 vektor. Elke komponent van die vektor stem ooreen met 'n funksie wat op sy bo- of ondergrens is. Gestel byvoorbeeld dat 'n probleem met vier funksies beskou word. Funksies 1, 2 en 3 is in die boom en funksie 4 is die tob-funksie. 'n Vektor met komponente $[1, 0, 1]$ beteken dan dat funksies 1 en 3 by hul bogrense en funksie 2 by sy ondergrens gestel moet word. Die ongebruikte deel van die hulpbron (indien daar hulpbron oor is) word aan funksie 4 toegedeel.

Laat die indeks t die tob-funksie aandui. Aanvanklik is $t = 0$. Laat $BG(i)$ en $OG(i)$ onderskeidelik die bo- en ondergrense van funksie i aandui. Laat $TBEG$ die hoeveelheid hulpbron wees wat oor is nadat 'n aantal van die funksies op hul bogrense toegeken is. Laat BEG die totale beskikbare hulpbron wees. Laat \underline{v} die binêre oplossingsvektor (die vektor waarmee die oplossing voorgestel word) wees en \underline{e}_j die binêre vektor waarvan alle komponente nul is behalwe die komponent in die j -de posisie wat 1 is.

Hier volg die stappe waarop die blaarsoekalgoritme berus:

Bereken die optimale toewysing van hulpbronne aan 'n aantal konvekse funksies oor wisselende intervalle:

1. [Verander die tob-funksie]
Laat $t = t + 1$. Vir $i = 1$ tot $n - 1$ laat $v_i = 0$. Stel $TBEG = BEG$.
2. [Bepaal 'n beginoplossing]
Stel die funksies (van regs in die binêre boom) een vir een vas by hul bogrense

totdat die hoeveelheid hulpbron opgebruik is.

Vir $i = n - 1$ tot 1

$TBEG = TBEG - BG(i)$

$v_i = 1$. Stop as $TBEG < BG(t)$ en ken $TBEG$ toe aan die tob-funksie. Gaan na Stap 4.

3. [Toets of daar nog van die hulpbron beskikbaar is of nie]

Indien $TBEG < BG(t)$, gaan na Stap 3a of anders gaan na Stap 3b.

 - (a) [Tel een by die oplossingsvektor]

$v = v \oplus e_{n-1}$.
 - (b) [Tel een by die laaste nie-nul element van die oplossingsvektor]

Gestel die laaste nie-nul element is die k -de komponent. Stel $v = v \oplus e_k$.
4. [Toets of 'n toelaatbare oplossing gegenerer is en, indien wel bereken die oplossing]

Bereken die tydelike doelfunksiewaarde indien die huidige oplossing 'n geldige oplossing is. Vervang die beste doelfunksiewaarde met die tydelike doelfunksiewaarde indien dit beter is as die beste doelfunksiewaarde.
5. [Bepaal of alle moontlike oplossings met die huidige tob-funksie al ondersoek is]

Indien $v_i = 1$ vir $i = 1, 2, 3, \dots, n - 1$, gaan na Stap 6; anders gaan na Stap 3.
6. [Bepaal of alle moontlike oplossings al ondersoek is] Indien $t < n$, gaan na Stap 1. Stop indien $t = n$.

Die pseudokode vir die blaarsoekalgoritme word in Algoritme 4.2 gegee. Die notasie wat daarin gebruik word, word in die hoofstuk oor notasie onder die afdeling oor gereserveerde simbole aan die begin van hierdie proefskrif gegee.

Algoritme 4.2 (Blaarsoekalgoritme)

Prosedure BLAARSOEK(HTP)

vir $i \leftarrow 1$ **tot** n **doen**

$B_t \leftarrow B$

$b_0 = 0$

$k \leftarrow 0$

vir $j \leftarrow 1$ **tot** a **doen**

as $j \neq i$ **dan**

$k = k + 1$

$I_t(k) \leftarrow j$

anders

$I_t(0) \leftarrow i$

end as

$b_j \leftarrow 0$

end vir

$D_t = 0$

▷ Elke funksie i word tob-funksie

▷ Kry 'n beginoplossing

```

vir  $j \leftarrow n - 1$  tot 1 doen Stap -1
   $B_t \leftarrow B_t - u_{I_t(j)}$ 
   $D_t \leftarrow D_t + f_{I_t(j)}(u_{I_t(j)})$ 
  as  $B_t < u_{I_t(0)}$  dan
     $B_t \leftarrow B_t + u_{I_t(j)}$ 
     $D_t \leftarrow D_t - f_{I_t(j)}(u_{I_t(j)})$ 
    Ontsnap vir
  anders
     $b_j \leftarrow 1$ 
  end as
end vir
as  $B_t > l_{I_t(0)}$  and  $B_t < u_{I_t(0)}$  dan
   $D_m \leftarrow D_t$ 
   $x_{I_t(j)} \leftarrow B_t$ 
  vir  $j \leftarrow 1$  tot  $n - 1$  doen
    as  $b_j = 1$  dan
       $x_{I_t(j)} \leftarrow u_{I_t(j)}$ 
    anders
       $x_{I_t(j)} \leftarrow l_{I_t(j)}$ 
    end as
  end vir
end as
herhaal
  herhaal
    as  $B_t \geq l_{I_t(0)}$  dan
       $\underline{b} \leftarrow \underline{b} \oplus e_{n-1}$ 
    anders
      vir  $j \leftarrow n - 1$  tot 1 doen Stap -1
        as  $b_j = 1$  dan
           $\underline{b} \leftarrow \underline{b} \oplus e_j$ 
        Ontsnap vir
      end as
    end vir
  end as
   $B_t \leftarrow B$ 
  as  $b_0 = 1$  dan
    Ontsnap herhaal
  end as
totdat  $B_t \geq l_{I_t(0)}$  en  $B_t \leq u_{I_t(0)}$ 
as  $b_0 = 1$  dan
  Ontsnap herhaal
end as
   $D_t \leftarrow f_{I_t(0)}(B_t)$ 
  vir  $j \leftarrow 1$  tot  $n - 1$  doen
    as  $b_j = 1$  dan

```



▷ Bereken tydelike oplossing


```

         $D_t \leftarrow D_t + f_{I_t(j)}(u_{I_t(j)})$ 
    anders
         $D_t \leftarrow D_t + f_{I_t(j)}(l_{I_t(j)})$ 
    end as
end vir
as  $D_t > D_m$  dan
     $D_m \leftarrow D_t$ 
     $x_{I_t(0)} \leftarrow B_t$ 
    vir  $j \leftarrow 1$  tot  $n - 1$  doen
        as  $b_j = 1$  dan
             $x_{I_t(j)} \leftarrow u_{I_t(j)}$ 
        anders
             $x_{I_t(j)} \leftarrow l_{I_t(j)}$ 
        end as
    end vir
end as
totdat  $b_0 = 1$ 
end vir
end Procedure

```

4.2.3 Taksoekalgoritme

In plaas daarvan om 'n hele tak van die oorsprong tot by die blaas as 'n oplossing te beskou, word elke nodus (vurk) as 'n oplossing beskou. Al die funksies verder af in die tak word op hul ondergrens vasgestel. Soos vir die blaarsoekalgoritme word elkeen van die funksies om die beurt as die tob-funksie beskou. Die filosofie agter hierdie algoritme is dat die soektog by die wortel van die boom begin. Indien daar nog van die hulpbron oorbly, word die volgende funksie op sy bogrens geplaas (die 1 word gekies by die vurk) en verminder die hulpbron met die bogrens van hierdie funksie. Hierdie proses word herhaal totdat daar net genoeg van die hulpbron oor is sodat die tob-funksie 'n toedeling tussen sy bo- en ondergrens kry. Nadat so 'n oplossing gevind is, word nog 'n funksie by sy bogrens bygevoeg. Drie situasies kan ontstaan tydens die ontvouing van die algoritme

1. Toelaatbare oplossing word gegenereer
Dit beteken die hulpbron word presies opgebruik deur die funksies by hul bogrense, of daar bly net genoeg oor sodat die res aan die tob-funksie toegedeel kan word. In hierdie geval word die doelfunksiewaarde bereken en nog 'n funksie word by sy bogrens bygevoeg.
2. Ontoelaatbare oplossing (met te min van die hulpbron)
In hierdie geval benodig die toedeling aan die funksies in die boom meer van die hulpbron as wat beskikbaar is. Die toedeling van die hulpbron word dus op 'n sistematiese wyse verander sodat een minder funksie in die boom op sy bogrens vasgestel word. In terme van die boom beteken dit dat daar van tak verander word

sodat 'n tak wat minder van die hulpbron gebruik, gekies word. Dit veroorsaak dat daar meer van die hulpbron beskikbaar raak, sodat 'n toelaatbare oplossing bereik kan word.

3. Ontoelaatbare oplossing (met te veel van die hulpbron)
Indien daar meer as die bogrens (van die tob-funksie) van die hulpbron oor is nadat die hulpbron aan elkeen van die funksies in die boom toegeken is, word daar dan op 'n sistematiese wyse een meer funksie wat op sy ondergrens vasgestel is, in die boom verander sodat dit op sy bogrens vasgestel word (daar word na 'n tak verander wat meer van die hulpbron gebruik). Dit veroorsaak dat daar minder van die hulpbron beskikbaar raak sodat 'n toelaatbare oplossing bereik kan word.

Indien alle moontlike toelaatbare oplossings bereik is, word die algoritme gestop. Dieselfde notasie as in die geval van die blaarsoekalgoritme word gebruik.

Laat $OOR(k)$ die maksimum hoeveelheid hulpbron wees wat aan die laaste $n - 1 - k$ funksie toegedeel kan word. Dit beteken dat indien $OOR(K) > \sum_{i=k}^{n-1}$, kan daar op geen manier 'n oplossing in daardie tak gegenereer word nie, omdat die oorblywende hulpbron (wat nog toegedeel moet word) meer is as die som van die funksies se bogrense waaraan nog van die hulpbron toegeken kan word.

Hieronder volg die stappe van die taksoekalgoritme:

Bereken die optimale (maksimum) toewysing van 'n hulpbron aan 'n aantal konvekse funksies oor wisselende intervalle.

1. [Verander die tob-funksie]
Laat $t = t + 1$. Vir $i = 1$ tot $n - 1$ laat $v_i = 0$. Stel $TBEG = BEG$.
2. [Bepaal 'n beginoplossing]
Stel die funksies (van regs in die binêre boom) een vir een by hul bogrense vas totdat die hoeveelheid tydelike hulpbron opgebruik is.
Vir $i = n - 1$ tot 1
 $TBEG = TBEG - BG(i)$
 $v_i = 1$. Stop as $TBEG < BG(t)$ en ken $TBEG$ aan die tob-funksie toe. Gaan na Stap 4.
3. [Toets of daar nog van die hulpbron beskikbaar is of nie]
Indien $TBEG < BG(t)$, gaan na Stap 3a. Indien $TBEG \geq BG(t)$ of $OOR(K) > \sum_{i=k}^{n-1}$, gaan na Stap 3b. Gestel die laaste nie-nul element is die k -de komponent van \underline{v} .
 - (a) [Stel nog een meer funksie by sy bogrens in die oplossingsvektor vas (beweeg af in die tak)]
 $\underline{v} = \underline{v} \oplus \underline{e}_{k+1}$.
 - (b) [Verminder die funksies op hul bogrense met een en stel 'n nuwe funksie vas op sy bogrens in die oplossingsvektor vas (vurk na 'n ander tak)]

Gestel die laaste nie-nul element is die k -de komponent dan $\underline{v} = \underline{v} \ominus \underline{e}_k$ en $\underline{v} = \underline{v} \oplus \underline{e}_{k-1}$.

4. [Toets of 'n toelaatbare oplossing gegenerer is en, indien wel bereken die oplossing] Bereken die tydelike doelfunksiewaarde indien die huidige oplossing 'n geldige oplossing is. Vervang die beste doelfunksiewaarde met die tydelike doelfunksiewaarde indien dit beter is as die beste doelfunksiewaarde.
5. [Bepaal of alle moontlike oplossings met die huidige tob-funksie al ondersoek is] Indien $v_i = 1$ vir $i = 1, 2, 3, \dots, n - 1$, gaan na Stap 6, anders gaan na Stap 3.
6. [Bepaal of alle moontlike oplossings al ondersoek is] Indien $t < n$, gaan na Stap 1. Stop indien $t = n$.

Die pseudokode vir die taksoekalgoritme word in Algoritme 4.3 verskaf.

Algoritme 4.3 (Taksoekalgoritme)

Prosedure TAKSOEK(HTP)

vir $i \leftarrow 1$ **tot** n **doen**

$R_s(0) \leftarrow R_s(0) + u_i$

end vir

vir $i \leftarrow 1$ **tot** n **doen**

$Klaar \leftarrow 0$

$B_t \leftarrow B$

$b_0 = 1$

$k \leftarrow 0$

vir $j \leftarrow 1$ **tot** a **doen**

as $j \neq i$ **dan**

$k = k + 1$

$I_t(k) \leftarrow j$

anders

$I_t(0) \leftarrow i$

end as

$b_j \leftarrow 0$

end vir

vir $j \leftarrow 1$ **tot** $n - 1$ **doen**

$B_t \leftarrow B_t - u_{I_t(j)}$

$b_j \leftarrow 1$

$k \leftarrow j$

as $B_t < u_{I_t(0)}$ **dan**

ontsnap vir

end as

end vir

herhaal

$j \leftarrow 1$



▷ Laat funksie i die tob-funksie wees

▷ Kry 'n beginoplossing

```

as  $B_t \geq u_{I_t(0)}$  dan
  as  $k < n - 1$  dan
    vir  $j \leftarrow k$  tot  $n - 1$  doen
       $b_j \leftarrow 1$ 
       $B_t \leftarrow B_t - u_{I_t(j)}$ 
       $k \leftarrow j$ 
      as  $B_t \leq u_{I_t(0)}$  dan
        ontsnap vir
      end as
    end vir
  anders
    vir  $j \leftarrow n - 1$  tot  $1$  doen Stap  $-1$ 
      as  $b_{j-1} = 1$  en  $b_j = 0$  dan
         $b_{j-1} \leftarrow 0$ 
         $b_j \leftarrow 1$ 
         $k \leftarrow j$ 
         $B_t \leftarrow B_t - u_{I_t(j)} + u_{I_t(j-1)}$ 
        ontsnap vir
      anders
        as  $b_j = 1$  dan
           $b_j \leftarrow 0$ 
           $B_t \leftarrow B_t + u_{I_t(j)}$ 
        end as
      end as
    end vir
  end as
anders as  $B_t \leq l_{I_t(0)}$  of  $B_t > OOR(K)$  dan
   $b_k \leftarrow 0$ 
   $B_t \leftarrow B_t + u_{I_t(k)}$ 
  vir  $j \leftarrow k$  tot  $0$  doen Stap  $-1$ 
    as  $b_j = 1$  dan
       $b_j \leftarrow 0$ 
       $b_{j+1} \leftarrow 1$ 
       $B_t \leftarrow B_t - u_{I_t(j)} + u_{I_t(j+1)}$ 
       $k \leftarrow j + 1$ 
    ontsnap vir
  end as
end vir
as  $B_t > u_{I_t(0)}$  dan
  gaan na TEVEEL
end as
end as
as  $B_t \geq l_{I_t(0)}$  en  $B_t \leq u_{I_t(0)}$  dan
   $D_t \leftarrow f_{I_t(0)}(B_t)$ 
  vir  $j \leftarrow 1$  tot  $b$  doen

```

```

    as  $b_j = 1$  dan
       $D_t \leftarrow D_t + f_{I_t(j)}(u_{I_t(j)})$ 
    anders
       $D_t \leftarrow D_t + f_{I_t(j)}(l_{I_t(j)})$ 
    end as
  end vir
as  $D_t > D_m$  dan
   $D_m \leftarrow D_t$ 
   $x_{I_t(0)} \leftarrow B_t$ 
  vir  $j \leftarrow 1$  tot  $n - 1$  doen
    as  $b_j = 1$  dan
       $x_{I_t(j)} \leftarrow u_{I_t(j)}$ 
    anders
       $x_{I_t(j)} \leftarrow l_{I_t(j)}$ 
    end as
  end vir
end as
as  $k < b$  dan
   $b_{k+1} \leftarrow 1$ 
   $k \leftarrow k + 1$ 
   $B_t \leftarrow B_t - u_{I_t(k)}$ 
anders
   $b_k \leftarrow 0$ 
   $B_t \leftarrow B_t + u_{I_t(k)}$ 
  vir  $j \leftarrow k - 1$  tot  $0$  doen Stap -1
    as  $b_j = 1$  dan
       $b_j \leftarrow 0$ 
       $b_{j+1} \leftarrow 1$ 
       $B_t \leftarrow B_t + u_{I_t(j)} - u_{I_t(j-1)}$ 
       $k \leftarrow j + 1$ 
    ontsnap vir
  end as
end vir
end as
TEVEEL:
  totdat  $b_0 = 0$ 
  end vir
end Prosedure

```



4.3 Samevatting

Presies soos in die vorige hoofstuk, bestaan die samevatting uit drie onderafdelings, naamlik 'n onderafdeling oor die metodologie, 'n onderafdeling met resultate en 'n onderafdeling waarin gevolgtrekkings verskaf word oor die resultate.

4.3.1 Metodologie

Daar bestaan nie 'n standaard datastel waarteen algoritmes wat HTP's oplos gemeet kan word nie en daarom word 'n stel probleme lukraak gegenereer om die nuwe algoritmes te toets. Die metodologie is soortgelyk aan die metodologie wat in §3.6.1 verskaf word. Die verskil tussen die metodologie in §3.6.1 en die metodologie in hierdie hoofstuk is dat Microsoft Excel se *randbetween*-funksie gebruik word om lukraak $u_i \in [100, 200]$ met $i = 1, 2, \dots, n$ en B tussen 0 en $\sum_{i=1}^n u_i$ te genereer.

Die kode vir hierdie algoritmes kan op die CD agterin hierdie proefskrif in die lêer met lêernaam `KEG.xls` gevind word.

In elke geval word daar 100 lukraak gegenereerde probleme opgelos. Dieselfde 100 lukraak gegenereerde probleme is met elkeen van die drie algoritmes opgelos en die berekeningstyd vasgelê.

4.3.2 Resultate

Die vergelykende resultate vir die blaarsoek-, taksoek- en vertak-en-begrens-algoritmes word in Tabel 4-1 gegee.

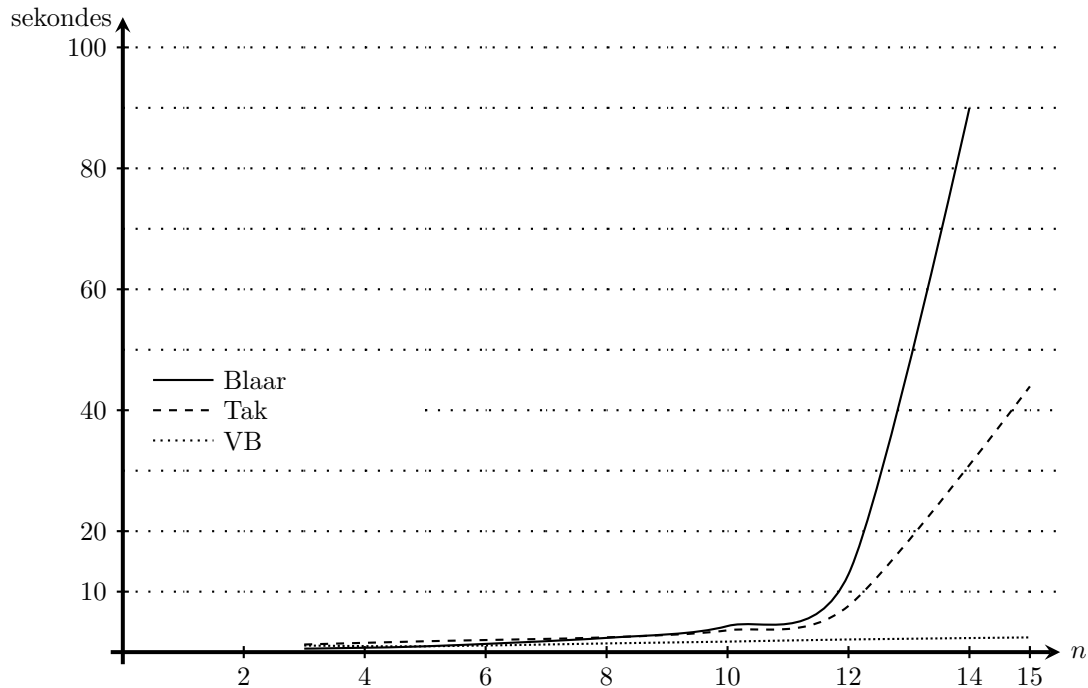


n	Blaar	Tak	VB
3	0.578	1.262	1.094
5	0.961	1.797	0.961
8	2.355	2.445	1.441
10	4.313	3.602	1.738
12	12.879	7.641	2.094
15	113.027	43.941	2.434

Tabel 4-1: Die oplossingstye in sekondes van 100 lukraak gegenereerde K/E/G-HTP's. Die volgende afkortings word in die tabel gebruik: *Blaar* = blaarsoekalgoritme, *Tak* = taksoekalgoritme, *VB* = vertak-en-begrens-algoritme.

Die resultate wat in Tabel 4-1 gegee, word ook grafies in Figuur 4.1 weer gegee.

Dit volg uit Tabel 4-1 en Figuur 4.1 dat die blaarsoek- en taksoekalgoritmes beduidend swakker vaar as die gewone vertak-en-begrens-algoritme. Tussen hierdie twee algoritmes vaar die taksoekalgoritme vir groter probleme (dit wil sê groter n) beter as die blaarsoekalgoritme. Die blaarsoekalgoritme is reeds vir klein probleme ($n = 15$) ± 50 maal en die taksoekalgoritme ± 20 maal stadiger as die gewone vertak-en-begrens-algoritme. In



Figuur 4.1: Grafiese voorstelling van die totale oplossingstye van drie verskillende algoritmes vir 100 $K/E/G$ -HTP's teenoor die aantal funksies n in die doelfunksie. Die volgende afkortings word gebruik: *Blaar* = blaarsoekalgoritme, *Tak* = taksoekalgoritme en *VB* = vertak-en-begrens-algoritme.

hierdie data is die probleme se grootte beperk tot $n = 15$. Die rede hiervoor is dat hierdie algoritmes etlike minute neem om 'n enkele groter probleem op te los, wat dit dus gewoon net te tydrowend maak om dit verder te ondersoek. Beide hierdie algoritmes se kompleksiteit neem so vinnig toe dat dit nie prakties gebruik kan word om groot $K/E/G$ -HTP's daarmee op te los nie.

4.3.3 Gevolgtrekkings

Die algoritmes wat in hierdie hoofstuk ondersoek is, is die logiese veralgemening van die breukalgoritme en die hellingalgoritme wat in Hoofstuk 3 ontwikkel is. Hierdie rigting van navorsing, naamlik om op versnelde maniere deur al die oplossings te soek wat aan die nodige voorwaardes van optimaliteit voldoen, blyk nie baie vrugbaar te wees nie. Hierdie tipe algoritmes is te berekeningsintensief, soos duidelik blyk uit die resultate in §4.3.2.

Hierdie hoofstuk se hoofbydrae is dus niks meer as om aan te toon dat soektogte deur al die oplossings wat aan die nodige voorwaardes voldoen nie 'n goeie strategie blyk te wees vir die oplossing van $K/E/G$ -HTP's nie. Hierdie tipe soektogte was egter wel suksesvol vir $K/E/V$ -HTP's waar beduidend vinniger algoritmes as byvoorbeeld vertak-en-begrens-algoritmes ontwikkel is.

Dit is nodig om daarop te let dat die blaarsoek- en taksoekalgoritmes eintlik baie nou verwant is aan die oplossing van 'n 0/1 knapsakprobleem. Dit is dan ook nie verbasend nie

dat hierdie twee algoritmes sterk ooreenkomste toon met die Horowitz-Shani-algoritme [24]. In wese is beide die blaar- en die taksoekalgoritmes eintlik net 'n aanpassing van implisiete enumerasie, wat gebruik kan word om 0/1 heeltallige programmeringsprobleme op te los.



Hoofstuk 5

$K/E/G$:

Vertak-en-begrens-algoritmes

Indien lineêre omsluitings saam met vertak-en-begrensing gebruik word om $K/E/G$ -HTP's op te los, kan daar in sekere gevalle oplossings met beter werklike doelfunksiewaardes gegenereer word indien meer beperkings by die probleem bygevoeg word namate daar in die takke van die vertak-en-begrensboom afbeweeg word. Daar moet dus in algoritmes wat vertak-en-begrensing gebruik baie seker gemaak word dat takke waarin daar later beter oplossings as die huidige kan wees, nie afgesny word nie. Een manier om hierdie probleem te oorkom, is om met 'n omsluiting (deur byvoorbeeld al die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$ met reguit lyne te benader) te werk. Die begrensing kan dan gedoen word deur die oplossing van die lineêre omsluiting met die werklike oplossing te vergelyk.



5.1 Inleiding tot vertak-en-begrens-algoritmes

In die algemeen berus 'n vertak-en-begrens-algoritme om nielineêre HTP's op te los daarop dat die lineêre omsluiting van die oorspronklike probleem opgelos word. Op grond van hierdie oplossing kan een van twee dinge gebeur, naamlik:

1. die oplossing van die lineêre omsluiting se doelfunksiewaarde is dieselfde as die doelfunksiewaarde van die oorspronklike probleem (wat optimaliteit van die oorspronklike probleem impliseer); of
2. indien die oplossing van die lineêre omsluiting se doelfunksiewaarde groter is as die doelfunksiewaarde van die oorspronklike probleem, word die toelaatbare gebied van die oorspronklike probleem in twee verdeel om twee subprobleme te vorm, wat elk opnuut omsluit word.

Tydens elke stap van die vertak-en-begrensing word daar dus óf geen óf twee subprobleme geskep. Die oorspronklike probleem vertak dus in twee subprobleme. Indien die twee subprobleme geskep word, word hierdie twee subprobleme die nuwe oorspronklike

probleme en word hulle lineêre omsluitings bereken en opgelos. Die beste optimale oplossing (ondergrens) van al die oorspronklike probleme word onthou en, mits 'n tak se lineêre omsluiting se doelfunksiewaarde minder as hierdie ondergrens word, kan daardie tak gestop word. Die proses word herhaal totdat alle takke gestop het.

Verder in hierdie hoofstuk word slegs *K/E/G*-HTP's beskou. Om in die proses van vertakking duidelik te kan kommunikeer, moet na verskillende probleme, subprobleme en versamelings van subprobleme verwys kan word. Hiervoor word nuwe terminologie eers informeel hieronder en in die volgende afdeling formeel ingevoer.

Noem die oorspronklike probleem die **ouerprobleem**. Indien die ouerprobleem vertak, vertak dit altyd in twee **kinders**, wat 'n binêre vertak-en-begrens-boom meebring. Indien die omsluiting met lyne gedoen word, kan aangetoon word dat die probleem altyd net in een plek kan vertak. Dit beteken dat, indien die vertakking plaasvind, die probleem altyd in twee subprobleme (**kinderprobleme**) gaan vertak. Dit gee dus aanleiding tot 'n binêre vertak-en-begrens-boom. Dit is belangrik dat die vertakking só moet geskied dat die twee kinderprobleme se toelaatbare gebiede onderling uitsluitend en alomvattend ten opsigte van die ouerprobleem se toelaatbare gebied moet wees. Die lineêre omsluiting van die twee kinderprobleme word dan geskep. Hierdie kinderprobleme word dan op hul beurt as nuwe ouerprobleme beskou wat weer, indien hulle vertak, hul eie kinderprobleme oplewer. Op hierdie manier word 'n hele nageslag van probleme uit die een oorspronklike probleem (**wortelprobleem**) geskep.

In die oplossing van die *K/E/G*-HTP beteken die vertakking dat die interval van waardes wat 'n veranderlike, sê x_r , kan aanneem in twee verdeel moet word. Gestel hierdie veranderlike kan in die ouerprobleem enige waarde in die interval $[l_r, u_r]$ aanneem. Gestel verder dat die interval verdeel moet word by die punt $x_r = R$, met $R \in (l_r, u_r)$. Dan sal x_r in die een kind enige waarde in die interval $[l_r, R]$ kan aanneem terwyl x_r in die ander kind enige waarde in die interval $[R, u_r]$ moet aanneem. Daar ontstaan dus altyd twee tipes kinders. Die een kind het vir al die veranderlikes presies dieselfde ondergrense as die ouer en die ander kind het vir al die veranderlikes presies dieselfde bogrense as die ouer. Daar word na hierdie twee kinders onderskeidelik as 'n **ondergrenskindprobleem** en 'n **bogrenskindprobleem** verwys. Gestel 'n ouerprobleem (H^o) word gegee deur

$$\text{maks} \quad \sum_{i=1}^n f_i(x_i^o)$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^n x_i^o &= B \\ x_i^o &\geq l_i && i = 1, 2, \dots, n \\ x_i^o &\leq u_i^o && i = 1, 2, \dots, n \\ x_i^o &\geq 0 && i = 1, 2, \dots, n, \end{aligned}$$

waar die boskrif o die ouerprobleem aandui. Laat die oplossing van die lineêre omsluiting van hierdie ouerprobleem gegee word deur $\underline{x}^o = \{u_1^o, u_2^o, \dots, u_{r-1}^o, R, l_{r+1}^o, \dots, l_n^o\}$, met

$R \in (l_r^o, u_r^o)$. Die ondergrenskindprobleem verskil van die ouerprobleem slegs in dié opsig dat die interval van waardes wat die veranderlike x_r kan aanneem, 'n ander bogrens het as in die ouerprobleem. Die ondergrenskindprobleem word gegee deur vir alle $i = 1, 2, \dots, r-1, r+1, r+2, \dots, n-1, n$

$$\begin{aligned} l_i^p &= l_i^o \\ u_i^p &= u_i^o \end{aligned}$$

en

$$\begin{aligned} l_r^p &= l_r^o \\ u_r^p &= R, \end{aligned}$$

waar die onderskrif p die p -de subprobleem (H_p) in die ondergrensnageslag voorstel. Die bogrenskindprobleem word op soortgelyk wyse gegee deur vir alle $i = 1, 2, \dots, r-1, r+1, r+2, \dots, n-1, n$

$$\begin{aligned} l_i^d &= l_i^o \\ u_i^d &= u_i^o \end{aligned}$$

en

$$\begin{aligned} l_r^d &= R \\ u_r^d &= u_r^o, \end{aligned}$$

waar die onderskrif d die d -de subprobleem (H_d) in die bogrensnageslag voorstel. Die funksies $f_i(x_i)$ bly dus presies dieselfde as in die oorspronklike probleem. Dit is net die grense van die definisieversameling van die r -de funksie wat in die twee subprobleme aangepas word.

In die algoritmes wat in hierdie hoofstuk volg, word daar dikwels lineêre omsluiting van sê die t -de subprobleem H_t opgelos eerder as H_t self. Hierdie lineêre omsluiting van H_t word voorgestel deur L_t en word gegee deur

$$\text{maks} \quad \sum_{i=1}^n c_i^t x_i^t$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^n x_i^t &= B^t \\ x_i^t &\geq l_i^t && i = 1, 2, \dots, n \\ x_i^t &\leq u_i^t && i = 1, 2, \dots, n \\ x_i^t &\geq 0 && i = 1, 2, \dots, n, \end{aligned}$$

waar

$$c_i^t = \frac{f_i(u_i^t) - f_i(l_i^t)}{u_i^t - l_i^t}.$$

Die vertakking by elke nodus na die ondergrenskindprobleem uit die ouerprobleem vorm dan die versameling van al die subprobleme wat presies dieselfde ondergrense het en word die ondergrensnageslag van 'n ouerprobleem. Die bogrensnageslag word op soortgelyke wyse uit al die bogrenskindprobleme gevorm. Al die bostaande begrippe word formeel hieronder gedefinieer.

5.2 Definisies

In hierdie afdeling volg 'n aantal nodige definisies om die subprobleme en takke in die vertak-en-begrens-boom te beskryf.

Definisie 5.1 (Ondergrensversameling)

Die ondergrensversameling \mathcal{G}_o van die t -de HTP H_t word gegee deur

$$\mathcal{G}_o = \{l_1^t, l_2^t, \dots, l_i^t, \dots, l_{n-1}^t, l_n^t\}.$$

Definisie 5.2 (Bogrensversameling)

Die bogrensversameling \mathcal{G}_b van H_t word gegee deur

$$\mathcal{G}_b = \{u_1^t, u_2^t, \dots, u_i^t, \dots, u_{n-1}^t, u_n^t\}.$$

Definisie 5.3 (Vertakking)

'n Vertakking in die vertak-en-begrens-algoritme om 'n $K/E/G$ -HTP op te los, is die skep van twee subprobleme uit die ouerprobleem sodat een veranderlike se interval in twee verdeel word. Dit beteken dat die twee subprobleme se toelaatbare gebiede onderling uitsluitend en alomvattend is ten opsigte van die ouerprobleem se toelaatbare gebied.

Definisie 5.4 (Ondergrenskindprobleem)

Die ondergrenskindprobleem (OGKP) is die subprobleem wat met 'n vertakking uit die ouerprobleem geskep word wat dieselfde ondergrensversameling, maar 'n ander bogrensversameling as die ouer het.

Definisie 5.5 (Bogrenskindprobleem)

Die bogrenskindprobleem (BGKP) is die subprobleem wat met 'n vertakking uit die ouer geskep word en wat dieselfde bogrensversameling, maar 'n ander ondergrensversameling as die ouer het.

Definisie 5.6 (Ondergrensnageslag)

Die ondergrensnageslag \mathcal{N}^{o,H_t} van 'n ouerprobleem, $\hat{s} H_t$, is die versameling van al die geslagte van OGKP's wat uit H_t vertak.

Die versameling \mathcal{N}^{o,H_t} bevat dus die hele nageslag van probleme wat uit H_t vertak en wat dieselfde \mathcal{G}_o as die ouerprobleem het. Die indeks van die algemene element in \mathcal{N}^{o,H_t} is k en daar is in totaal K elemente in die versameling. Dit is

$$\mathcal{N}^{o,H_t} = \{N_1^{o,H_t}, N_2^{o,H_t}, \dots, N_{k-1}^{o,H_t}, N_k^{o,H_t}, N_{k+1}^{o,H_t}, \dots, N_K^{o,H_t}\}.$$

Definisie 5.7 (Bogrensnageslag)

Die bogrensnageslag \mathcal{N}^{b,H_t} van 'n ouerprobleem, sê H_t , is die versameling van al die geslagte van BGKP's wat uit daardie ouerprobleem vertak.

Die versameling \mathcal{N}^{b,H_t} bevat dus die hele nageslag van probleme wat uit 'n ouerprobleem vertak en wat dieselfde \mathcal{G}_b as die ouerprobleem het. Die indeks van die algemene element in \mathcal{N}^{b,H_t} is m en daar is in totaal M elemente in die versameling. Dit is

$$\mathcal{N}^{b,H_t} = \{N_1^{b,H_t}, N_2^{b,H_t}, \dots, N_{m-1}^{b,H_t}, N_m^{b,H_t}, N_{m+1}^{b,H_t}, \dots, N_M^{b,H_t}\}.$$

Definisie 5.8 (Iso-boversameling) Die iso-boversameling (\mathcal{I}^b) bevat al die BGKP's wat uit die ondergrensnageslag van alle subprobleme vertak en word geskryf as

$$\mathcal{I}^b = \{H_1^b, H_2^b, \dots, H_d^b, \dots, H_{D-1}^b, H_D^b\}.$$

Definisie 5.9 (Iso-ondersameling) Die iso-ondersameling (\mathcal{I}^o) bevat al die OGKP's wat uit die ondergrensnageslag van alle subprobleme vertak en word geskryf as

$$\mathcal{I}^o = \{H_1^o, H_2^o, \dots, H_p^o, \dots, H_{P-1}^o, H_P^o\}.$$

In al die gevalle waar die iso-bo- en iso-ondersamelings gebruik word, is daar dus sprake van twee tipes subprobleme. Dus moet daar in hierdie gevalle nog 'n boskryf bygevoeg word om na byvoorbeeld die ondergrens van die i -de funksie van H_p^o te verwys. Hierdie ondergrens sal dus gegee word deur $l_i^{o,p}$.

Om die begrensing te doen, word twee tipes doelfunksies gebruik, naamlik die doelfunksiewaarde van die lineêre omsluiting en die doelfunksiewaarde van die werklike onderliggende probleem. Laat $\underline{x}^t = \{x_1^t, x_2^t, \dots, x_i^t, \dots, x_n^t\}$ 'n oplossing vir die t -de subprobleem van $K/E/G$ -HTP wees. Laat $\underline{c}^t = \{c_1^t, c_2^t, \dots, c_i^t, \dots, c_n^t\}$ die kostekoëffisiënte (hellings) van die lyne in die doelfunksie wees. Laat D^b die beste werklike doelfunksiewaarde uit die eerste $t - 1$ subprobleme wees en \underline{x}^b die oplossing wat daarmee ooreenstem. Die werklike doelfunksiewaarde word in Definisie 5.10 gegee.

Definisie 5.10 (Werklike doelfunksiewaarde)

Indien die lineêre omsluiting van die t -de subprobleem se oplossing gegee word deur $\underline{x}^t = (x_1^t, x_2^t, \dots, x_n^t)$, word die doelfunksiewaarde $D_w(\underline{x}^t)$ van die werklike onderliggende probleem gegee deur

$$D_w(\underline{x}^t) = \sum_{i=1}^n f_i(x_i^t).$$

Gestel die t -de subprobleem se i -de veranderlike strek oor die interval $[l_i^t, u_i^t]$, met $l_i \leq l_i^t < l_i^t \leq u_i^t \leq u_i$.

Definisie 5.11 (Lyndoelfunksiewaarde)

Laat $\underline{x}^t = (x_1^t, x_2^t, \dots, x_n^t)$ 'n oplossing vir die lineêre omsluiting van die t -de subprobleem wees, dan word die lyndoelfunksiewaarde $D_L(t)$ gegee deur

$$D_L(\underline{x}^t) = \sum_{i=1}^n (f_i(l_i^t) + c_i^t(x_i^t - l_i^t))$$

met

$$c_i^t = \frac{f_i(u_i^t) - f_i(l_i^t)}{u_i^t - l_i^t}.$$

Die lyndoelfunksiewaarde is dus niks anders as die doelfunksiewaarde indien die oplossing \underline{x}^t in L^t ingestel word nie.

Dit volg vir enige subprobleem t vanuit die definisie van konveksiteit en die feit dat al die funksies $f_i(x_i)$ konveks en stygend is dat op enige interval, $[l_i^t, u_i^t]$, met $l_i \leq l_i^t < u_i^t \leq u_i$, die lyn wat die koördinate $(l_i^t, f_i(l_i^t))$ en $(u_i^t, f_i(u_i^t))$ verbind, altyd groter of gelyk aan die funksiewaarde $f_i(x_i)$, met $x_i \in [l_i^t, u_i^t]$, sal wees. Hierdie feit kan gebruik word om die begrensing vir die vertak-en-begrens-algoritmes te doen. Gestel die beste doelfunksiewaarde tot voor subprobleem t is D^b . Indien vir subprobleem t met optimale oplossing \underline{x}^t volg dat

$$D^b \geq D_L(\underline{x}^t), \tag{5.1}$$

kan daardie tak gestaak word. Vir alle moontlike oplossings vir subprobleem t volg dat

$$D_w(\underline{x}^t) \leq D_L(\underline{x}^t). \tag{5.2}$$

Uit (5.1) en (5.2) volg dat vir alle moontlike oplossings van subprobleem t sal

$$D^b \geq D_L(\underline{x}^t) \geq D_w(\underline{x}^t).$$

Dus sal geen nageslag wat uit subprobleem t voortspruit tot beter oplossings as D^b in staat wees nie. Dit beteken dat, indien $D^b \geq D_L(\underline{x}^t)$, alle vertakkings uit subprobleem t gestaak kan word.

5.3 Isogrens-vertak-en-begrens-benadering

Die isogrens-vertak-en-begrens-benadering werk op die beginsel dat 'n subversameling van subprobleme van 'n probleem opgelos word en nie net een subprobleem soos wat die geval is met gewone vertak-en-begrens-algoritmes nie.

5.3.1 Isogrensheuristiek

Die isogrensheuristiek bestaan bloot daaruit dat die iso-bogrensprosedure en die iso-ondergrensprosedure uitgevoer word. Die beste oplossing wat in die twee prosedures bereken is, is die antwoord van die isogrensheuristiek.

Heuristiek 5.1 (Isogrens)

Roep: Iso-ondergrensprosedure

Roep: Iso-bogrensprosedure

Afvoer: D^b en \underline{x}^b

Daar is gevalle wanneer die isogrensheuristiek die optimale oplossing vind. In sommige gevalle is dit selfs moontlik om vir seker te sê dat die isogrensheuristiek se oplossing wel die optimale oplossing is. Dit gebeur indien die iso-bogrensprosedure en die iso-ondergrensprosedure dieselfde oplossing gee. Die enigste manier waarop die iso-bogrensprosedure en die iso-ondergrensprosedure dieselfde oplossing kan lewer, is as daar geen subprobleme deur enige van die twee geskep word nie, wat impliseer dat die twee prosedures die hele vertak-en-begrens-boom ondersoek het.

Die iso-ondergrens- en die iso-bogrensprosedure word in die volgende twee subafdelings beskryf.

5.3.1.1 Iso-ondergrensprosedure

Die iso-ondergrensprosedure is afgelei van die primale simpleksalgoritme met bogrense vir die oplossing van 'n lineêre HTP. Soos met die oplossing van die lineêre probleem, begin die iso-ondergrens algoritme ook om die veranderlike x_i wat met die funksie $f_i(x_i)$ ooreenstem en die hoogste gemiddelde helling het, te komplementeer. (Sien §2.10 vir 'n verduideliking van die begrip komplementering van 'n veranderlike.) Hierdie proses van komplementering word herhaal in volgorde van hoogste tot laagste kostekoëffisiënt (gemiddelde helling) totdat die volgende veranderlike, sê x_p , se komplementering sou veroorsaak dat die regterkant van die beperking negatief word. Op hierdie punt word die bogrens van hierdie veranderlike aangepas sodat die komplementering veroorsaak dat die regterkant van die beperking nul word. Dit beteken $u_p = b$. Aangesien 'n verandering in bogrens 'n verandering in die kostekoëffisiënt meebring, moet c_p ook aangepas word, omdat die gemiddelde helling verander. Dit kan beteken dat 'n ander funksie nou 'n kandidaat word om gekomplementeer te word. Sou hierdie nuwe veranderlike se komplementering ook veroorsaak dat die regterkant van die beperking negatief word, word hierdie veranderlike se bogrens ook aangepas na b . Hierdie proses word herhaal totdat 'n veranderlike gekomplementeer word wat die regterkant van die vergelyking in die tablo nul maak. Elke keer dat die bogrens van 'n veranderlike aangepas word, word daar effektief 'n ondergrenskind van die betrokke probleem geskep. Aangesien daar nooit aan die ondergrens van 'n veranderlike verander word nie, word die volle ondergrensnageslag van die betrokke probleem deur die proses van komplementering geskep.

Algoritme 2.4 moet aangepas word om elke keer die lyn van die tob-veranderlike¹ x_r wat 'n toedeling $x_r \in (l_r, u_r)$ kry, aan te pas na die lyn tussen $(l_r, f_r(l_r))$ en $(x_r, f_r(x_r))$. Hierdie aanpassing moet voortgaan totdat al die funksies 'n toedeling gelyk aan hul onder- en/of bogrense kry. Hierdie proses kan gedoen word deur 'n ekstra stap by die iso-ondergrensprosedure by te voeg.

Let daarop dat, aangesien 'n veranderlike se bogrens aangepas word indien hy 'n kandidaat vir 'n simpleksiterasie is, 'n simpleksiterasie nooit in die iso-ondergrensprosedure uitgevoer sal word nie. Optimaliteit van die ondergrensnageslag kan dus altyd met behulp van slegs komplementering verseker word.

'n Volledige lys van die notasie wat gebruik word, word aan die begin van hierdie proefskrif verskaf. Die pseudokode vir die iso-grensprosedures word in Prosedure 5.1 gegee.

Prosedure 5.1 (Iso-ondergrens)

Doel: Bepaal die optimale oplossing vir die ondergrensnageslag van 'n $K/E/G$ -HTP.

Stap 1

▷ Inisieer

Bereken die kostekoeffisiënte met $c_i^0 = \frac{f_i(u_i)}{u_i}$
 $u_i^0 \leftarrow u_i$ vir alle $i = 1, 2, \dots, n$
 $k \leftarrow 0$

Stap 2

as $B^k = 0$ dan

Doen Stap 5 en **Stop**

anders

Bereken $D_L(\underline{x}^k)$ en $D_w(\underline{x}^k)$.

as $D_L(\underline{x}^k) \leq D^b$ dan

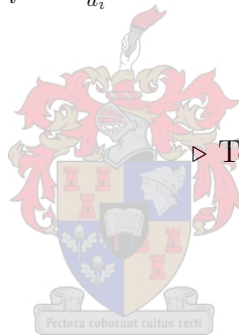
Gaan na Stap 6

anders as $D_w(\underline{x}^k) > D^b$ dan

Gaan na Stap 5

end as

end as



▷ Toets vir begrensing en optimaliteit

Stap 3

▷ Bepaal niebasisveranderlike om aan te pas of te komplementeer

$z_r^k \leftarrow \min_i [z_i^k]$ vir alle $i = 1, 2, \dots, n$ gegewe dat $z_i^k < 0$.²

Stap 4

▷ Komplementeer 'n veranderlike of pas 'n bogrens aan en vertak

as $u_r^k \leq B^k$ dan

▷ Komplementeer veranderlike x_r

$B^k \leftarrow B^k - u_r^k$

$c_r^k \leftarrow -c_r^k$

$z_r^k \leftarrow -z_r^k$

Gaan na Stap 2

¹tussen-onder-en-bogrens-veranderlike

²Dit is dieselfde as om $c_r^k = \max_i [c_i^k]$ vir alle i sodat $z_i^k < 0$ te bepaal.

anders as $u_r^k > B^k$ **dan** ▷ Pas veranderlike x_r se bogrens aan (skep OGKP)

$$k \leftarrow k + 1$$

$$u_r^k \leftarrow B^{k-1}$$

$$c_r^k \leftarrow \frac{f_r(u_r^m) - f_r(l_r^m)}{u_r^m - l_r^m}$$

$$z_r^k \leftarrow c_r^k$$

Gaan na Stap 3

end as

Stap 5 ▷ Pas beste doelfunksie en oplossing aan
 Laat $D^b \leftarrow D_w(\underline{x}^k)$ en $\underline{x}^b \leftarrow \underline{x}^k$. Gaan na Stap 2.

Stap 6 ▷ Tak is begrens
Stop

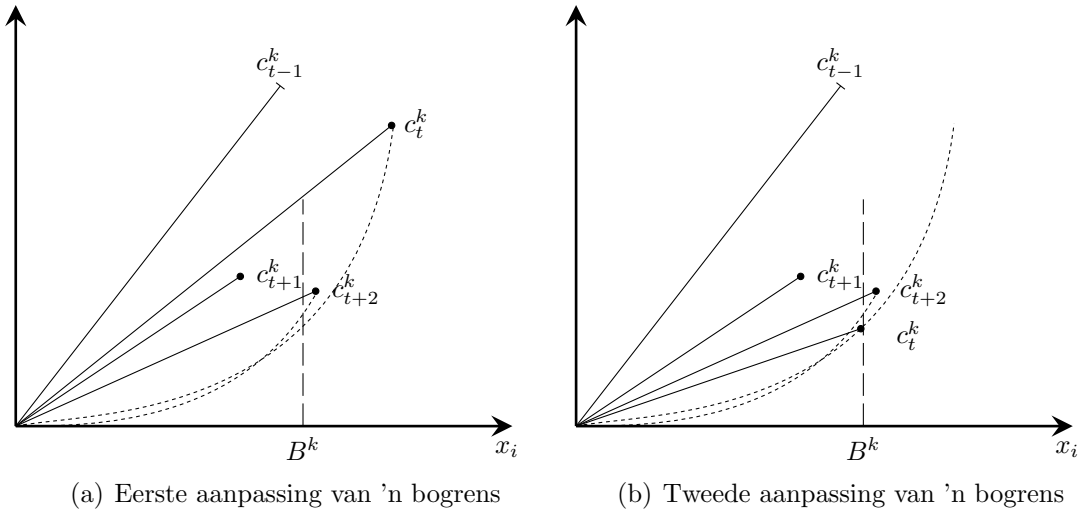
Die iso-ondergrensprosedure lewer die optimale oplossing van die ondergrensnageslag van 'n probleem.

Die werking van Prosedure 5.1 kan soos volg geïllustreer word: In Figuur 5.1 word die konvensie gebruik dat die lineêre omsluiting van 'n veranderlike wat 'n toedeling gelyk aan sy bogrens gekry het, met 'n dwars strepie gemerk word en een wat nog geen toedeling gekry het nie, met 'n kolletjie.³ Aanvaar sonder verlies van algemeenheid dat die funksies $f_i(x_i)$ in volgorde van afnemende c_i^k , die lineêre omsluiting van 'n funksie $f_i(x_i)$ se helling, gerangskik is. Gestel x_{t-1} is die laaste veranderlike wat in Prosedure 5.1 gekomplementeer is en dat x_t die eersvolgende kandidaat vir komplementering is. Die probleem kan dan soos in Figuur 5.1(a) lyk. Die regterkant B^k van die beperking is nou kleiner as die bogrens van die volgende kandidaat x_t , wat gekomplementeer moet word. Dit beteken dat veranderlike x_t se bogrens verander (verminder) moet word na B^k [Stap 4]. Die probleem lyk nou soos in Figuur 5.1(b), met x_{t+1} die veranderlike met die grootste c_i^k . Komplementeer hierdie veranderlike omdat $u_{t+1}^k < B^k$ en skep subprobleem $k+1$ [Stap 4]. Dit beteken dat die regterkant van die beperking met u_{t+1} , die bogrens van veranderlike x_{t+1} , verminder, oftewel $B^{k+1} = B^k - u_{t+1}^k$.

Uit die versameling veranderlikes wat nog nie by hul bogrense⁴ is nie, het x_{t+2} die hoogste gemiddelde helling c_i^{k+2} . Hierdie veranderlike se bogrens is groter as die huidige regterkant B^{k+1} van die beperking en die bogrens moet dus aangepas word [Stap 4]. Die situasie is nou soos in Figuur 5.2(a). Uit die veranderlikes wat nog nie by hul bogrens is nie, is die een met die grootste c_i^{k+1} die veranderlike x_t . Ook x_t se bogrens is groter as B^{k+1} en daarom moet x_t ook gekomplementeer word [Stap 4]. Die veranderlike x_t bly steeds die een met die hoogste c_i^{k+1} uit die veranderlikes wat nog nie 'n toedeling gelyk aan hul bogrens gekry het nie. Dus word veranderlike x_t gekomplementeer [Stap 4]. Die regterkant van die beperking B^{k+2} is nou nul, wat beteken die algoritme moet stop.

³Dit beteken byvoorbeeld in Figuur 5.1(a) dat die veranderlike x_{t-1} 'n toedeling gelyk aan sy bogrens gekry het en dat al die ander veranderlikes (x_r tot x_{t+2}) 'n toedeling gelyk aan hul ondergrens het.

⁴Dit wil sê wat tans gelyk aan hul ondergrense is.



Figuur 5.1: Grafiese voorstelling van die werking van die iso-ondergrensprosedure.

Die iso-ondergrensprosedure waarborg dat die finale toedeling só geskied dat die funksie wat 'n toedeling tussen sy bo- en ondergrense kry, die grootste toename in doelfunksiewaarde sal lewer uit die versameling van funksies wat op daardie stadium steeds 'n toedeling gelyk aan hul ondergrense het. Anders gestel beteken dit dat hierdie prosedure verseker dat die tob-funksie optimaal bepaal word uit die oorblywende funksies, wat nog geen toedeling ontvang het nie.

Dit is duidelik dat elke keer wanneer die bogrens van 'n veranderlike aangepas word, die nuwe ondergrenskind van daardie probleem in effek geskep word.

5.3.1.2 Iso-bogrensprosedure

Die iso-bogrensprosedure is afgelei van die duale simpleksalgoritme met bogrense. Die werking daarvan stem sterk ooreen met die iso-ondergrensprosedure, behalwe dat hierdie prosedure elke keer die ondergrens groter maak indien die komplementering van 'n veranderlike veroorsaak dat die regterkant van die beperking positief gaan word. In hierdie geval word daar weer nooit aan die bogrense van veranderlikes verander nie, wat tot gevolg het dat die bogrensnageslag van 'n probleem geskep word.

Die pseudokode vir die iso-bogrensprosedure word in Prosedure 5.2 gegee. Die notasie wat in Prosedure 5.2 gebruik word, kan aan die begin van hierdie proefskrif gevind word.

Prosedure 5.2 (Iso-bogrens)

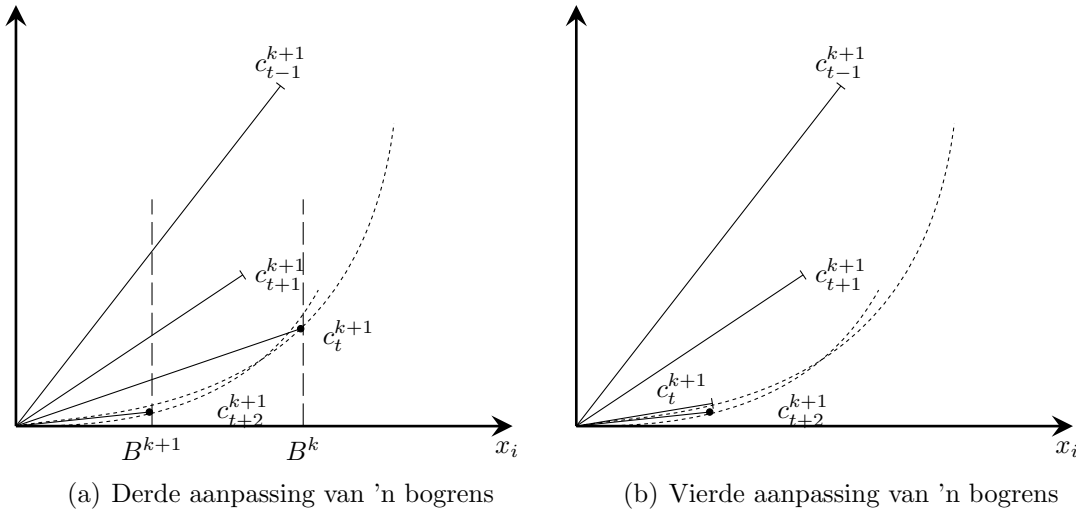
Doel: Bepaal die optimale oplossing vir die bogrensnageslag van 'n $K/E/G$ -HTP.

Stap 1

▷ Inisieer

Doen die lineêre transformasie $x_i = x'_i - l_i$.

Bereken die kostekoëffisiënte met $c_i^0 = \frac{f_i(u_i)}{u_i}$



Figuur 5.2: Grafiese voorstelling van die werking van die iso-ondergrensprosedure (vervolg...).

$$u_i^0 \leftarrow u_i \text{ vir alle } i = 1, 2, \dots, n$$

$$m \leftarrow 0$$

Stap 2

as $B^m = 0$ dan

Doen Stap 5 en **Stop**

anders

Bereken $D_L(\underline{x}^m)$ en $D_w(\underline{x}^m)$.

as $D_L(\underline{x}^m) \leq D^b$ dan

Gaan na Stap 6

anders as $D_w(\underline{x}^m) > D^b$ dan

Gaan na Stap 5

end as

end as



▷ Toets vir begrensing en optimaliteit

Stap 3

▷ Bepaal niebasisveranderlike om aan te pas of te komplementeer

$z_r^m \leftarrow \max_i [z_i^m]$ vir alle $i = 1, 2, \dots, n$ gegewe dat $z_i^m < 0$.⁵

Stap 4

▷ Komplementeer 'n veranderlike of pas 'n bogrens aan en vertak

as $u_r^m \leq B^m$ dan

▷ Komplementeer veranderlike x_r

$B^m \leftarrow B^m - u_r^m$

$c_r^m \leftarrow -c_r^m$

$z_r^m \leftarrow -z_r^m$

Gaan na Stap 2

⁵Dit is dieselfde as om $c_r^m = \min_i [c_i^m]$ vir alle i sodat $z_i^m < 0$ te bepaal.

anders as $u_r^m > B^m$ **dan** \triangleright Pas veranderlike x_r se ondergrens aan (skep 'n BGKP)

$$m \leftarrow m + 1$$

$$l_i^m \leftarrow l_i^{m-1} \text{ vir } i = 1, 2, \dots, n$$

$$l_r^m \leftarrow u^{m-1} - B^{m-1}$$

$$c_r^m \leftarrow \frac{f_r(u_r^m) - f_r(l_r^m)}{u_r^m - l_r^m}$$

$$\text{Lineêre transformasie: } x_r = x_r' - l_r^m$$

$$z_r^m \leftarrow -c_r^m$$

Gaan na Stap 3

end as

Stap 5 \triangleright Pas beste doelfunksie en oplossing aan

Laat $D^b \leftarrow D_w(\underline{x}^m)$ en $\underline{x}^b \leftarrow \underline{x}^m$. Gaan na Stap 2.

Stap 6 \triangleright Vertakking is begrens

Stop

Die werking van die iso-bogrensprosedure is presies soortgelyk aan die werking van die iso-ondergrensprosedure. Die enigste verskil is dat die ondergrens elke keer aangepas word en al die veranderlikes aanvanklik gekomplementeerd is. Dit wil sê daar word aanvanklik meer as die hulpbron B gebruik. Die veranderlikes word dan van die kleinste c_i 's af wêr gekomplementeerd.

Die iso-bogrensprosedure waarborg dat die finale aftrekking so geskied dat die funksie wat 'n toedeling tussen sy bo- en ondergrense kry, die grootste toename in doelfunksiewaarde sal lewer uit die versameling van funksies wat op daardie stadium steeds 'n toedeling gelyk aan hul bogrense het. Anders gestel beteken dit dat hierdie prosedure verseker dat die tob-funksie optimaal uit die oorblywende funksies, wat reeds 'n toedeling gelyk aan hul bogrense ontvang het, bepaal word.

5.3.1.3 $K/E/V$ -HTP's en die isogrensheuristiek

In die spesiale geval waar $K/E/V$ -HTP's met die isogrensheuristiek opgelos word, kan optimaliteit van die heuristiek se oplossing gewaarborg word.

Stelling 5.1

Die isogrensheuristiek bepaal die globale optimum vir $K/E/V$ -HTP's

Bewys: Al wat bewys moet word, is dat alle moontlike oplossings deur óf die iso-ondergrensprosedure óf die iso-bogrensprosedure ondersoek word, want dit sou impliseer dat die globale optimum dan gevind sal wees. Indien die hoeveelheid hulpbron 'n veelvoud van die vaste interval van die veranderlikes is, sal die optimale oplossing deur beide die iso-ondergrensprosedure en die iso-bogrensprosedure gevind word. Gestel dus die hoeveelheid hulpbron B is nie 'n veelvoud van die vaste interval van die veranderlikes nie.

Dit beteken die optimale tob-funksie moet bepaal word. Die iso-ondergrensprosedure begin deur alle veranderlikes gelyk aan hul ondergrense te stel. Die iso-ondergrensprosedure komplementeer, dit is stel toedeling aan die veranderlikes gelyk aan hul bogrense, vanaf die veranderlike met die hoogste gemiddelde helling totdat die volgende veranderlike se komplementering 'n negatiewe regterkant in die simplekstablo gaan veroorsaak. Gestel hierdie veranderlike is x_r , met $1 \leq r \leq n$. Wat die prosedure hiervandaan (van x_r tot x_n) doen, is om elkeen van hierdie veranderlikes as 'n kandidaat vir komplementering te beskou. Die veranderlike word dan óf gekomplementeer óf die bogrens word aangepas totdat die komplementering uiteindelik 'n regterkant van 0 in die simplekstablo verseker. Wat in werklikheid gebeur, is dat die iso-ondergrensheuristiek soek na die beste tob-funksie uit die laaste $n - r$ funksies; met ander woorde die beste tob-funksie uit die funksies $f_r(x_r), f_{r+1}(x_{r+1}), \dots, f_{n-1}(x_{n-1}), f_n(x_n)$ word bepaal. Volgens presies dieselfde argumente vir die duale simpleksalgoritme met bogrense volg dat die iso-bogrensprosedure die beste tob-funksie uit die eerste r funksies oftewel uit funksies $f_1(x_1), f_2(x_2), \dots, f_{r-1}(x_{r-1}), f_r(x_r)$, kies. Dit wil sê die iso-ondergrens- en iso-bogrensprosedure saam beskou elke funksie as 'n kandidaat vir die tob-funksie. Indien in ag geneem word dat die tob-funksie net een moontlike waarde kan aanneem, beteken dit alle moontlike oplossings wat aan die nodige voorwaardes voldoen, is deursoek. Die globale optimum moet een van hierdie punte wees. ■

5.3.2 Isogrens-vertak-en-begrens-algoritme

In hierdie afdeling word die isogrens-vertak-en-begrens-algoritme voorgestel. Dit is in wese ook 'n vertak-en-begrens-algoritme, maar dit is meer effektief as die normale vertak-en-begrens-benadering wat in Algoritme 2.1 voorgestel word. In werklikheid word elke subprobleem deur beide die iso-ondergrens- en die iso-bogrensalgoritme opgelos, terwyl gesorg word dat daar nie duplisering plaasvind nie. Die beginsel onderliggend aan die isogrens-vertak-en-begrens-algoritme is om tydens een iterasie die probleem te vertak in vier subversamelings van subprobleme. Die eerste subversameling is die ondergrensnageslag (\mathcal{N}^{o, H_t}) van die betrokke probleem (of subprobleem), die tweede is die bogrensnageslag (\mathcal{N}^{b, H_t}) van dieselfde probleem (of subprobleem). Die derde versameling is die iso-boversameling (\mathcal{I}^b) wat uit probleme bestaan wat nog met behulp van die iso-bogrensalgoritme opgelos moet word. Die vierde versameling is die iso-onderversameling (\mathcal{I}^o) wat uit probleme bestaan wat nog met behulp van die iso-ondergrensalgoritme opgelos moet word. Die iso-boversameling vertak uit die ondergrensnageslag en die iso-onderversameling vertak uit die bogrensnageslag. In een iterasie word daar dus in plaas van een subprobleem, twee subversamelings – die ondergrensnageslag en bogrensnageslag – van daardie subprobleem opgelos.

Die drie algoritmes wat hieronder volg, moet eintlik as een gesien word aangesien die isogrens-, iso-ondergrens- en iso-bogrensalgoritmes eintlik net drie prosedures van dieselfde algoritme is. Anders gestel beteken dit dat die isogrens-vertak-en-begrens-algoritme die ander twee algoritmes roep om elke keer 'n subversameling van subprobleme op te los.

Die isogrens-vertak-en-begrens-algoritme maak gebruik van die iso-ondergrensalgoritme⁶ en iso-bogrensalgoritme⁷ wat hieronder gegee word. Die ondergrens- en die bogrensalgoritme moet dus eintlik as subroetines gesien word wat deur die isogrens-vertak-en-begrens-algoritme gebruik word.

Hieronder volg 'n beskrywing van die stappe van die isogrens-vertak-en-begrens-algoritme.

1. [Inisieer]

Stel al die indekse en veranderlikes se waardes vas en skep die eerste probleem in \mathcal{I}^o om op te los.
2. [Maak 'n siklus wat al die subprobleme in \mathcal{I}^o en \mathcal{I}^b oplos]
 - (a) [Los subprobleem p op met behulp van ondergrensalgoritme]

Roep die iso-ondergrensalgoritme om subprobleem $p \in \mathcal{I}^o$ op te los.
 - (b) [Los subprobleem d op met behulp van bogrensalgoritme]

Roep die iso-bogrensalgoritme om subprobleem $d \in \mathcal{I}^b$ op te los.
3. [Uitset: Optimale oplossing]

Die optimale doelfunksiewaarde is D^b met ooreenstemmende oplossing \underline{x}^b , soos wat in beide die ondergrens- en bogrensalgoritmes bygewerk is. Stop.

Vervolgens word die pseudokode van die isogrens-vertak-en-begrens-algoritme in Algoritme 5.1 gegee.

Algoritme 5.1 (Isogrens)

Bereken die optimale toewysing van hulpbronne aan 'n aantal konvekse funksies oor wisselende intervalle.

Doel: Bepaal die optimale oplossing vir 'n gegewe $K/E/G$ -HTP deur van Algoritmes 5.2 en 5.3 gebruik te maak.

Stap 1

▷ Inisieer

$P \leftarrow 1$
 $D \leftarrow 0$
 $D^b \leftarrow 0$
 $d \leftarrow 0$
 $p \leftarrow 1$
 $T^s \leftarrow 0$
 $B^1 \leftarrow B$

⁶Die iso-ondergrensalgoritme is vir alle praktiese doeleindes dieselfde as die iso-ondergrensprosedure wat in die isogrensheuristiek gebruik word. Die enigste verskil is dat hierdie algoritme ook ekstra subprobleme skep wat nog deur die iso-bogrensalgoritme opgelos moet word.

⁷Dieselfde ooreenkomste as dié tussen die iso-ondergrensalgoritme en iso-ondergrensprosedure, bestaan hier tussen die iso-bogrensalgoritme en die iso-bogrensprosedure. Die enigste verskil is dat die iso-bogrensalgoritme ekstra subprobleme skep wat nog deur die iso-ondergrensalgoritme opgelos moet word.

$\underline{x}^b \leftarrow (0, 0, \dots, 0)$
vir $i \leftarrow 1$ **tot** n **doen** ▷ Skep die eerste probleem in \mathcal{I}^o
 $c_i^1 = \frac{f_i(u_i) - f_i(l_i)}{u_i - l_i}$
 $u_i^1 \leftarrow u_i$
 $l_i^1 \leftarrow l_i$
end vir

Stap 2 ▷ Skep 'n siklus wat alle probleme in \mathcal{I}^o en \mathcal{I}^b oplos

terwyl $T_s < 1$ **doen**
as $p < P$ **dan**
 $p \leftarrow p + 1$
Roep: Iso-ondergrensalgoritme(H_p^o)
end as
as $d < D$ **dan**
 $d \leftarrow d + 1$
Roep: Iso-bogrensalgoritme(H_d^b)
end as
as $p < P$ **dan**
 $T_s \leftarrow 0$
anders as $d < D$ **dan**
 $T_s \leftarrow 0$
anders
 $T_s \leftarrow 1$
end as
end terwyl



Stap 3 ▷ Beste antwoord is bereik
Uitset: D^b
Uitset: \underline{x}^b .
Stop

Hier volg 'n beskrywing van die stappe van die iso-ondergrensalgoritme:

Laat k die indeks wees oor die ondergrensnageslag (\mathcal{N}^{o, H_t}) wat van (sub)probleem t geskep word. Die voetskrif k saam met 'n veranderlike dui dus die k -de ondergrenskind in \mathcal{N}^{o, H_t} aan.

1. [Toets vir begrensing en optimaliteit]
 Bereken lyn- en werklike doelfunksiewaardes. Indien die regterkant van die beperking nul is, is die optimale oplossing vir die subprobleem gevind. Gaan na Stap 4. Indien die lyndoelfunksie kleiner as die beste doelfunksiewaarde is, gaan na Stap 5. Andersins, indien die werklike doelfunksiewaarde beter is as die beste doelfunksiewaarde, gaan na Stap 4.
2. [Bepaal niebasisveranderlike om aan te pas of te komplementeer]
 Bepaal die kleinste inskrywing uit die negatiewe inskrywings in die z -ry van die

simplekstablo. Die veranderlike wat met hierdie kolom ooreenstem, is die kandidaat vir komplementering.

3. [Komplementeer 'n veranderlike of pas 'n bogrens aan en vertak]

Gaan na Stap 3a indien die bogrens van die gekose veranderlike kleiner of gelyk is aan die regterkant van die beperking. Gaan na Stap 3b indien die bogrens van die veranderlike groter as die regterkant van die beperking is.

 - (a) [Komplementeer]

Komplementeer die gekose veranderlike. Gaan na Stap 1.
 - (b) [Vertak en pas bogrens aan]

Vertak die probleem in die gekose veranderlike.

Skep die BGKP: Die BGKP se ondergrens word gelyk gestel aan die regterkant van die beperking. Voeg hierdie subprobleem by die iso-boversameling (\mathcal{I}^b).

Skep die OGKP: Pas die bogrens van die gekose veranderlike aan na die regterkant van die beperking. Bereken die gekose veranderlike se nuwe koëffisiënt en die nuwe waardes in die z ry.

Gaan na Stap 1.
4. [Pas beste doelfunksie en oplossing aan]

Werk die beste doelfunksiewaarde en ooreenstemmende oplossing by. Gaan na Stap 1.
5. [Tak is begrens]

Stop



Vervolgens word die pseudokode vir die iso-ondergrensalgoritme in Algoritme 5.2 gegee.

Algoritme 5.2 (Iso-ondergrens)

Doel: Bepaal die optimale oplossing vir die ondergrensnageslag van 'n gegewe *K/E/G*-HTP en skep die subprobleme wat uit die gegewe probleem vertak in die vertak-en-begrensing.

$k \leftarrow 0$
 $D \leftarrow 0$

Stap 1

▷ Toets vir begrensing en optimaliteit

as $B^k = 0$ dan

Gaan na Stap 4

Stop

anders

Bereken $D_L(\underline{x}^k)$ en $D_w(\underline{x}^k)$.

as $D_L(\underline{x}^k) \leq D^b$ dan

Gaan na Stap 5

anders as $D_w(\underline{x}^k) > D^b$ dan

Gaan na Stap 4
 end as
 end as

Stap 2 \triangleright Bepaal niebasisveranderlike om aan te pas of te komplementeer
 $z_r^k \leftarrow \min_i [z_i^k]$ vir alle $i = 1, 2, \dots, n$ gegewe dat $z_i^k < 0$.

Stap 3 \triangleright Komplementeer 'n veranderlike of vertak
 as $u_r^k \leq B^k$ dan \triangleright Komplementeer veranderlike x_r

$$B^k \leftarrow B^k - b_r^k$$

$$c_r^k \leftarrow -c_r^k$$

$$z_r^k \leftarrow -z_r^k$$

Gaan na Stap 1

anders as $u_r^k > B^k$ dan \triangleright Vertak

$$k \leftarrow k + 1$$

$$u_r^k \leftarrow B^{k-1}$$

$$c_r^k \leftarrow \frac{f_r(u_r^k) - f_r(l_r^k)}{u_r^k - l_r^k}$$

$$z_r^k \leftarrow -c_r^k$$

\triangleright Skep OGKP (N_k^{o, H_p^o}) en plaas dit in \mathcal{N}^{o, H_p^o}

$$D \leftarrow D + 1$$

vir $i \leftarrow 1$ tot n doen

$$l_i^{b, D} \leftarrow l_i^k$$

$$u_i^{b, D} \leftarrow u_i^k$$

$$c_i^{b, D} \leftarrow c_i^k$$

end vir

$$l_r^{b, D} \leftarrow B^{k-1}$$

$$B^{b, D} \leftarrow B$$

$$c_r^{b, D} \leftarrow \frac{f_r(u_r^{b, D}) - f_r(l_r^{b, D})}{u_r^{b, D} - l_r^{b, D}}$$

$$\mathcal{I}^b \leftarrow \mathcal{I}^b \cup H_D^b$$

Gaan na Stap 1

end as



\triangleright Skep BGKP (H_D^b) en plaas dit in \mathcal{I}^b

Stap 4 \triangleright Pas beste doelfunksie en oplossing aan
 Laat $D^b \leftarrow D_w(k)$ en $\underline{x}^b \leftarrow \underline{x}^k$. Gaan na Stap 1.

Stap 5 \triangleright Vertakking is begrens
 Stop \triangleright Gaan terug na Algoritme 5.1

Die iso-bogrens algoritme stem weer ooreen met die iso-bogrensprosedure en bepaal dus ook die beste oplossing uit die bogrensnageslag van 'n gegewe probleem en subprobleem. Elke keer wanneer 'n ondergrens aangepas word, word 'n vertakking gemaak (subprobleem geskep) wat by die iso-ondersameling (\mathcal{I}^o) bygevoeg word. Alle probleme in \mathcal{I}^o moet nog deur die ondergrensalgoritme opgelos word.

'n Kort beskrywing van wat elke stap in die iso-bogrens algoritme doen, volg hieronder:

Laat $m = 0$, waar m 'n teller is van die aantal kinders in die bogrensnageslag wat geskep is.

1. [Toets vir begrensing en optimaliteit]

Bereken lyn- en werklike doelfunksiewaardes. Indien die regterkant van die beperking nul is, is die optimale oplossing vir die subprobleem gevind. Gaan na Stap 4. Indien die lyndoelfunksie kleiner as die beste doelfunksiewaarde is, gaan na Stap 5, anders, indien die werklike doelfunksiewaarde beter as die beste doelfunksiewaarde is, gaan na Stap 4.
2. [Bepaal niebasisveranderlike om aan te pas of te komplementeer]

Bepaal die kleinste positiewe inskrywing in die z -ry van die simplekstablo. Die veranderlike wat met hierdie kolom ooreenstem, is die kandidaat vir komplementering.
3. [Komplementeer 'n veranderlike of pas 'n ondergrens aan en vertak]

Gaan na Stap 3a indien die gekose veranderlike se bogrens groter of gelyk is aan die negatief van die regterkant van die beperking. Gaan na Stap 3b indien die gekose veranderlike se bogrens kleiner as die negatief van die regterkant van die beperking is.

 - (a) [Komplementeer]

Komplementeer die gekose veranderlike. Gaan na Stap 1.
 - (b) [Vertak en pas ondergrens aan]

Skep die OGKP: Vertak die probleem by die gekose veranderlike sodat die gekose veranderlike se bogrens gelyk is aan die negatief van die regterkant van die beperking. Voeg hierdie subprobleem by die iso-onderversameling (\mathcal{I}^o).

Skep die BGKP: Pas die gekose veranderlike se ondergrens aan na sy bogrens minus die regterkant van die beperking en bereken die nuwe koëffisiënt en z -ry.

Gaan na Stap 1.
4. [Pas beste doelfunksie en oplossing aan]

Werk die beste doelfunksiewaarde en ooreenstemmende oplossing by. Gaan na Stap 1.
5. [Tak is begrens]

Stop

Die pseudokode vir hierdie algoritme word in Algoritme 5.3 gegee.

Algoritme 5.3 (Iso-bogrens)

Doel: Bepaal die optimale oplossing vir die ondergrensnageslag van 'n gegewe *K/E/G*-HTP en skep die subprobleme wat uit die gegewe probleem vertak in die vertak-en-begrensing.

$m \leftarrow 0$
 $D \leftarrow 0$

Stap 1

as $B^m = 0$ dan

Gaan na Stap 4

Stop

anders

Bereken $D_L(\underline{x}^m)$ en $D_w(\underline{x}^m)$.

as $D_L(\underline{x}^m) \leq D^b$ dan

Gaan na Stap 5

anders as $D_w(\underline{x}^m) > D^b$ dan

Gaan na Stap 4

end as

end as

▷ Toets vir begrensing en optimaliteit

Stap 2

▷ Bepaal niebasisveranderlike om aan te pas of te komplementeer

$z_r^m \leftarrow \min_i [z_i^m]$ vir alle $i = 1, 2, \dots, n$ gegewe dat $z_i^m > 0$.

Stap 3

as $u_r^m \leq B^m$ dan

$B^m \leftarrow B^m - b_r^m$

$c_r^m \leftarrow -c_r^m$

$z_r^m \leftarrow -z_r^m$

Gaan na Stap 1

anders as $u_r^m > B^m$ dan

$m \leftarrow m + 1$

$l_r^m \leftarrow B^{m-1}$

$c_r^m \leftarrow \frac{f_r(u_r^m) - f_r(l_r^m)}{u_r^m - l_r^m}$

$z_r^m \leftarrow -c_r^m$

$P \leftarrow P + 1$

vir $i \leftarrow 1$ tot n doen

$l_i^{o,P} \leftarrow l_i^m$

$u_i^{o,P} \leftarrow u_i^m$

$c_i^{o,P} \leftarrow c_i^m$

end vir

$u_r^{o,P} \leftarrow B^{m-1}$

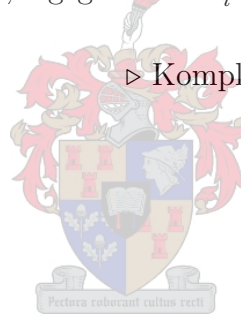
$B^{o,P} \leftarrow B$

$c_r^{o,P} \leftarrow \frac{f_r(u_r^{o,P}) - f_r(l_r^{o,P})}{u_r^{o,P} - l_r^{o,P}}$

$\mathcal{I}^b \leftarrow \mathcal{I}^b \cup H_P^b$

Gaan na Stap 2

end as



▷ Komplementeer 'n veranderlike of vertak

▷ Komplementeer veranderlike x_r

▷ Vertak

▷ Skep BGKP (N_m^{b,H_d^b}) en plaas dit in \mathcal{N}^{b,H_d^b}

▷ Skep OGKP en plaas dit in \mathcal{I}^o

Stap 4

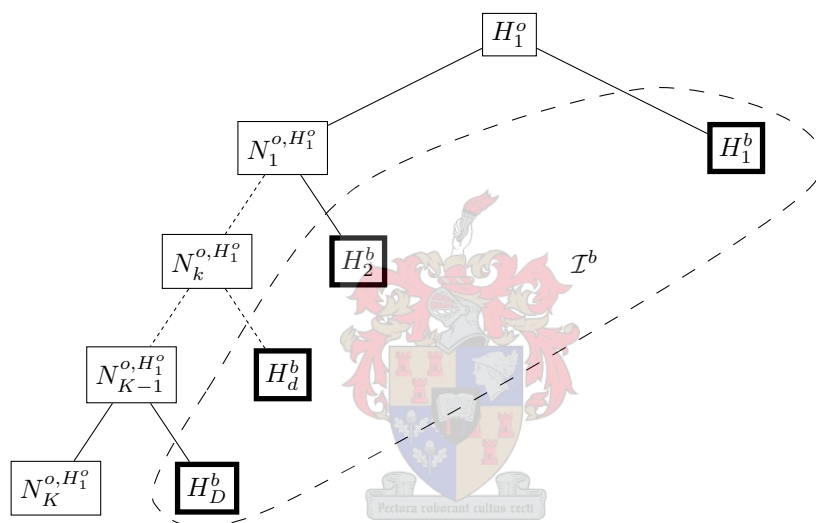
▷ Pas beste doelfunksie en oplossing aan

Laat $D^b \leftarrow D_w(\underline{x}^m)$ en $\underline{x}^b \leftarrow \underline{x}^m$. Gaan na Stap 1.**Stap 5****Stop**

▷ Vertakking is begrens

▷ Gaan terug na Algoritme 5.1

Die werking van die isogrens-vertak-en-begrens-algoritme kom dus in wese daarop neer dat dit met die lineêre omsluiting van die wortelprobleem begin en dan in Stap 2 (Roep: iso-ondergrensalgoritme) die optimale oplossing vir die ondergrensnageslag daarvan en terselfdertyd al die vertakkings uit hierdie subversameling bepaal. Die vertak-en-begrensboom lyk dan soos in Figuur 5.3.

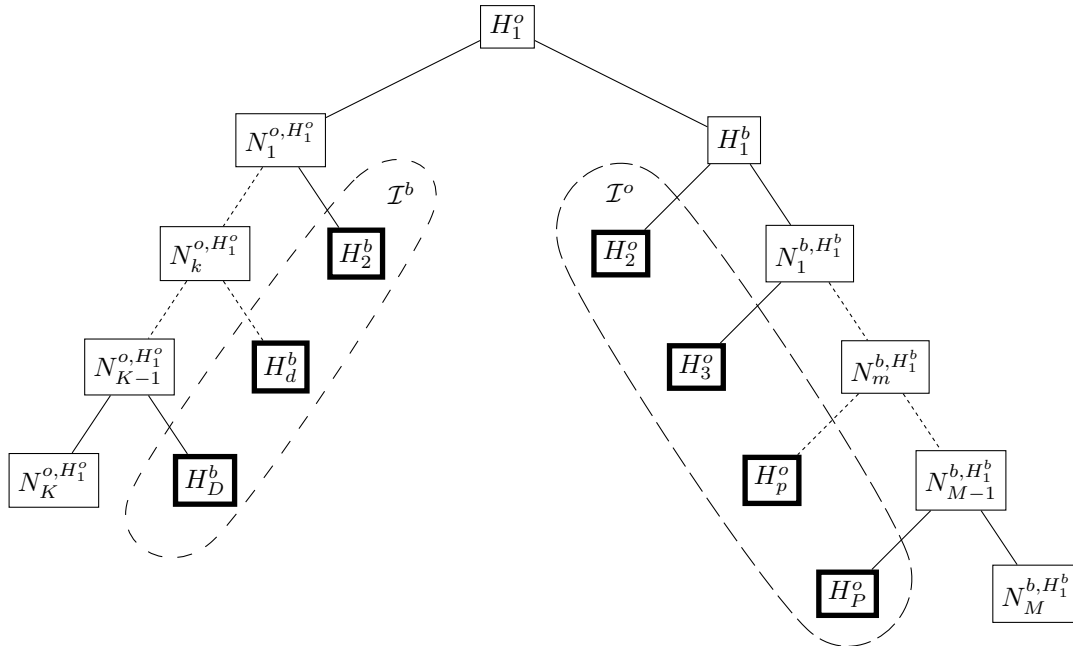


Figuur 5.3: Vertak-en-begrensboom na Stap 2 (Roep: iso-ondergrensalgoritme) van die isogrens-vertak-en-begrens-algoritme. Die nodusse met die dik lyne verteenwoordig subprobleme wat nog opgelos moet word. In hierdie stadium is al hierdie probleme in \mathcal{T}^b .

Die eerste bogrenskindprobleem (die eerste vertakking wat in Stap 2 (Roep: iso-ondergrensalgoritme) geskep is) word in Stap 2 (Roep: iso-bogrensalgoritme) opgelos. Indien hierdie stap afgehandel is, lyk die vertak-en-begrensboom soos die een in Figuur 5.4.

5.4 Vertak-sny-en-begrens-algoritme

Soms kan 'n gedeelte van die toelaatbare gebied gedurende die vertakking afgesny word. In hierdie gevalle kan 'n gedeelte van (of selfs die hele) interval van die ouerprobleem se tob-funksie in die ondergrenskind afgesny word. Net so kan 'n gedeelte van (of die hele) interval van die ouerprobleem se bogrenskind afgesny word. Die rede waarom dele van die tob-funksie se interval afgesny kan word, is omdat die optimale oplossing vir die



Figuur 5.4: Vertak-en-begrens-boom na Stap 2 (Roep: iso-bogrens algoritme) van die iso-ondergens algoritme. Die nodusse met die dik lyne verteenwoordig subprobleme wat nog opgelos moet word.

subprobleem wat hierdie afgesnyde deel (of dele) van die toelaatbare gebied bevat, reeds bepaal is.

Voordat die werking van die algoritme beskryf kan word, is die volgende definisies nodig. Beskou die $K/E/G$ -HTP. Sonder verlies van algemeenheid kan aanvaar word dat vir $i = 1, 2, \dots, n$ al die funksies $f_i(x_i)$ in volgorde van afnemende gemiddelde helling gesorteer is. 'n Definisie vir die ondersnit volg hieronder.

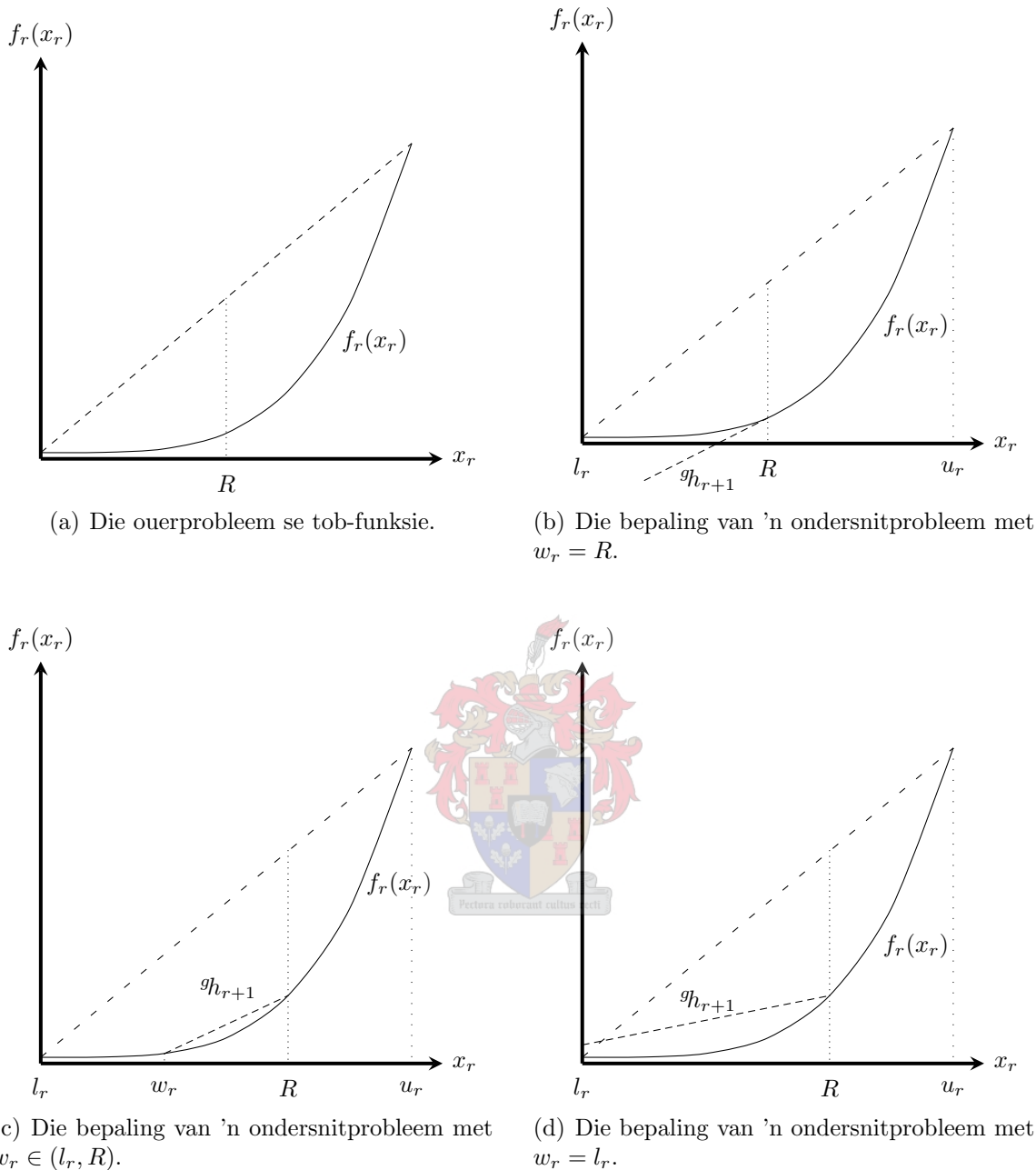
Definisie 5.12 (Ondersnit)

Die ondersnit vir 'n funksie $f_r(x_r)$ by die punt $R \in [l_r, u_r]$ is die gedeelte van die funksie $f_r(x_r)$ tussen die grense $[v_r, R]$, waar

$$v_r = \text{maks} \left(l_r; \min_x \{x : f_r(x) = g_{h_{r+1}}x + f_r(R) - R g_{h_{r+1}}\} \right).$$

Die waardes wat x kan aanneem, kan nooit groter as R wees nie, omdat die funksie $f_r(x)$ altyd gelyk aan $g_{h_{r+1}}x + f_i(R) - R g_{h_{r+1}}$ vir $x = R$ sal wees. Hieruit en uit Definisie 5.12 is dit duidelik dat een van drie situasies kan ontstaan:

1. Die kleinste waarde van x waarvoor die gelykheid $f_r(x) = g_{h_{r+1}}x + f_r(R) - R g_{h_{r+1}}$ geld, is $x = R$. Dit beteken dat die ondersnit slegs in een punt bestaan, oftewel in die interval $[R, R]$. Hierdie geval word skematies in Figuur 5.5(b) aangetoon.



Figuur 5.5: Die bepaling van 'n ondersnit.

2. Die kleinste waarde van x waarvoor die gelykheid $f_r(x) = g_{h_{r+1}}x + f_r(R) - Rg_{h_{r+1}}$ geld, is $l_r < x = w_r < R$. Dit beteken dat die ondersnit oor die interval $[w_r, R]$ strek. Hierdie geval word in Figuur 5.5(c) getoon.
3. Die kleinste waarde van x waarvoor die gelykheid $f_r(x) = g_{h_{r+1}}x + f_r(R) - Rg_{h_{r+1}}$ geld, is $x < l_r$. Dit beteken dat die ondersnit oor die interval $[l_r, R]$ bestaan. Hierdie geval word skematies in Figuur 5.5(d) gegee.

Beskou 'n optimale oplossing van die lineêre omsluiting van 'n $K/E/G$ -HTP. Gestel hierdie optimale oplossing het 'n tob-funksie by $i = r$ (anders is die optimale oplossing vir die omsluiting ook die optimale oplossing van die HTP) met 'n toedeling van $R \in (l_r, u_r)$ aan die tob-funksie. Dit beteken dat, volgens die vertak-en-begrens-metode, die HTP in die punt $x_r = R$ moet vertak. Die ondersnitprobleem kan nou gedefinieer word.

Definisie 5.13 (Ondersnitprobleem)

Die ondersnitprobleem is daardie kindprobleem van die probleem of subprobleem in die vertak-en-begrensing waarin die definisieversameling van die tob-funksie $f_r(x_r)$ slegs oor die ondersnit strek.

Die bo-snit kan op soortgelyke wyse as die ondersnit gedefinieer word.

Definisie 5.14 (Bo-snit)

Die bo-snit vir 'n funksie $f_r(x_r)$ by die punt $R \in [l_r, u_r]$ is die gedeelte van die funksie $f_r(x_r)$ tussen die grense $[R, w_r]$, waar

$$w_r = \min \left(u_r; \quad \max_x \{x : f_r(x) = {}^g h_{r-1}x + f_r(R) - R {}^g h_{r-1}\} \right).$$

Die waardes wat x kan aanneem, kan nooit kleiner as R wees nie, omdat die funksie $f_r(x)$ altyd gelyk aan ${}^g h_{r-1}x + f_r(R) - R {}^g h_{r-1}$ sal wees vir $x = R$. Hierdie feit saam met Definisie 5.14 kan gebruik word om by 'n soortgelyke drie situasies as wat die geval met die ondersnit was, uit te kom.

1. Die grootste waarde van x waarvoor die gelykheid $f_r(x) = {}^g h_{r-1}x + f_r(R) - R {}^g h_{r-1}$ geld, is $x = R$. Dit beteken dat die bo-snit slegs in een punt bestaan, oftewel in die interval $[R, R]$. Hierdie situasie is soortgelyk aan die situasie wat in Figuur 5.5(b) getoon word.
2. Die grootste waarde van x waarvoor die gelykheid $f_r(x) = {}^g h_{r-1}x + f_r(R) - R {}^g h_{r-1}$ geld, is $R < x = w_r < u_r$. Dit beteken dat die bo-snit oor die interval $[R, w_r]$ strek. Hierdie situasie is soortgelyk aan die situasie wat in Figuur 5.5(c) getoon word.
3. Die grootste waarde van x waarvoor die gelykheid $f_r(x) = {}^g h_{r-1}x + f_r(R) - R {}^g h_{r-1}$ geld, is $x > u_r$. Dit beteken dat die bo-snit oor die interval $[R, u_r]$ bestaan. Hierdie situasie is soortgelyk aan die situasie wat in Figuur 5.5(d) getoon word.

Die bo-snitprobleem volg dan ook op soortgelyke wyse as die ondersnitprobleem.

Definisie 5.15 (Bo-snitprobleem (BSP))

Die bo-snitprobleem is die kindprobleem van 'n probleem of subprobleem in die vertak-en-begrensing waarin die definisieversameling van die tob-funksie $f_r(x_r)$ slegs oor die bo-snit strek.

Beskou, soos tevore, 'n optimale oplossing van die lineêre omsluiting van 'n $K/E/G$ -HTP. Gestel hierdie optimale oplossing het 'n tob-funksie by $i = r$ (anders is die optimale

oplossing vir die omsluiting ook die optimale oplossing van die HTP) met 'n toedeling van $R \in (l_r, u_r)$ aan die tob-funksie. Dit beteken dat, volgens die vertak-en-begrens-metode, die HTP in die punt $x_r = R$ moet vertak. Gestel in die ondersnitprobleem strek die tob-funksie se definisieversameling oor die interval $[v_r, R]$. Dan kan die ondersnitkindprobleem soos hieronder gedefinieer word.

Definisie 5.16 (Ondersnitkindprobleem (OSKP))

Die ondersnitkindprobleem van 'n $K/E/G$ -HTP is die subprobleem ná die vertakking waarin die tob-funksie vervang word met dieselfde funksie, maar wat net oor die interval $[l_r, v_r]$ strek.

Dit beteken dat die ondersnitkindprobleem verenig met die ondersnitprobleem (OSP) die ondergrenskindprobleem sal wees, oftewel

$$\text{OSKP} \cup \text{OSP} = \text{OGKP}.$$

Die definisie van die bo-snitkindprobleem volg presies soortgelyk aan die definisie van die ondersnitkindprobleem.

Definisie 5.17 (Bo-snitkindprobleem (BSKP))

Die bo-snitkindprobleem van $K/E/G$ -HTP is die subprobleem ná die vertakking waarin die tob-funksie slegs oor die interval $[w_r, u_r]$ strek.

Dit beteken dat die bo-snitkindprobleem verenig met die bo-snitprobleem (BSP) die bogrenskindprobleem sal wees, oftewel

$$\text{BSKP} \cup \text{BSP} = \text{BGKP}.$$

Definisie 5.18 (Snitprobleem (SP))

Die snitprobleem is die vereniging van die bo-snitprobleem en die ondersnitprobleem.

Stelling 5.2 (Optimaliteit van die ondersnitprobleem)

Gestel die funksies $f_i(x_i)$ is gerangskik van grootste tot kleinste gemiddelde helling. Laat 'n optimale oplossing vir die lineêre omsluiting van $K/E/G$ gegee word deur $\underline{x} = (u_1, u_2, \dots, u_{r-1}, w, l_{r+1}, \dots, l_n)$, met $w \in (l_r, u_r)$. Dan sal \underline{x} ook 'n optimale oplossing wees vir die subprobleem waarin die tob-funksie vervang word met sy ondersnit by w .

Bewys: Indien die ondersnit net in 'n punt bestaan, dit wil sê in die subprobleem bestaan die tob-funksie $f_r(x_r)$ slegs oor die interval $[w, w]$, is die bewys triviaal. Gestel dus dat die tob-funksie in die subprobleem oor 'n interval strek. Gestel die snit bestaan oor die interval $x_r \in [v, w]$ met $l_r \leq v < w$. Dan gaan die gemiddelde helling (lineêre omsluiting) van die ondersnit gelyk wees aan die gemiddelde helling van $f_{r+1}(x_{r+1})$. Indien die primale algoritme met bogrense (Algoritme 2.4) gebruik word, sal die funksie in volgorde van

afnemende gemiddelde helling gekomplementeer (dus op hul bogrense gestel) word. Die eerste $r - 1$ funksies sal dus gekomplementeer word. Die regterkant van die beperking sal dan gelyk wees aan w , wat beteken die funksie met die volgende hoogste helling moet 'n toedeling van w kry om optimaliteit te bewerkstellig. Uit die definisie van die ondersnit sal die funksies $f_r(x_r)$ en $f_{r+1}(x_{r+1})$ dieselfde hellings hê en ook die volgende hoogste helling wees. Kies dan altyd funksie $f_r(x_r)$ waaraan die w toegedeel kan word, maar dan is die ondersnit optimaal en die oplossing is presies gelyk aan \underline{x} . Dus sal \underline{x} , wat die optimale oplossing vir die lineêre omsluiting was, ook die optimale oplossing vir die ondersnit wees. ■

Stelling 5.3 (Optimaliteit van die bo-snitprobleem)

Laat die optimale oplossing vir lineêre omsluiting van $K/E/G$ gegee word deur $\underline{x} = (u_1, u_2, \dots, u_{r-1}, x_r, l_{r+1}, \dots, l_n)$, dan sal \underline{x} die optimale oplossing wees vir die subprobleem waarin die tob-funksie met die bogrenssnit vervang word.

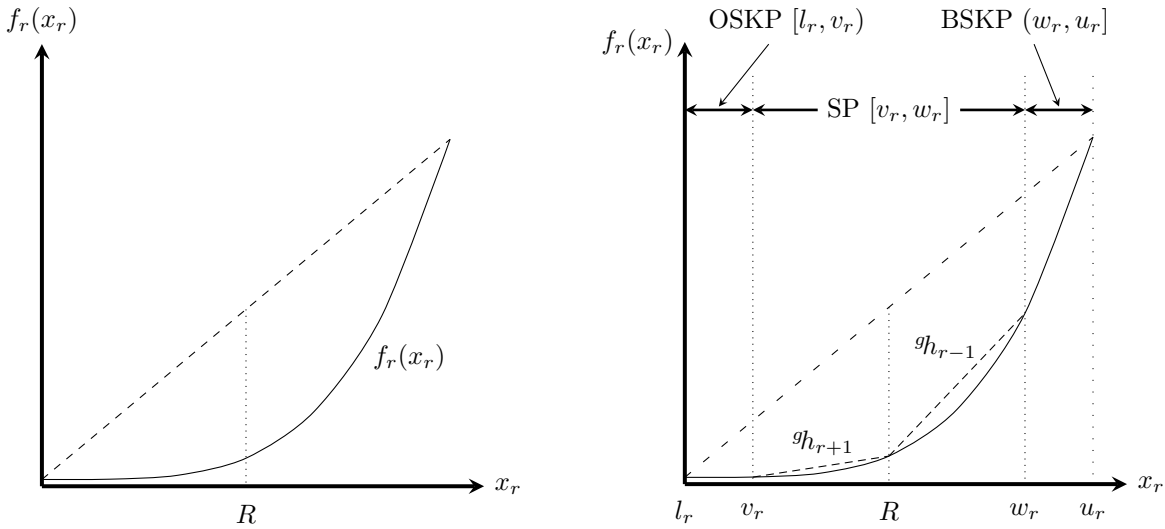
Die bewys van hierdie stelling volg met presies dieselfde argumente as in die bewys van Stelling 5.2. Die enigste verskil is dat die bo-snit en die duale algoritme met bogrense (Algoritme 2.5) gebruik word. In hierdie geval sal funksies $f_r(x_r)$ en $f_{r-1}(x_{r-1})$ weer dieselfde hellings hê. In die duale algoritme met bogrense kan enige van hierdie funksies dan op w vasgestel word. Kies altyd $f_r(x_r)$ vir hierdie vasstelling sodat die oplossing dieselfde as \underline{x} is.

Uit Stellings 5.2 en 5.3 volg die volgende afleiding. Gestel die funksies $f_i(x_i)$ is gerangskik van grootste tot kleinste gemiddelde helling.

Afleiding 5.1 (Optimaliteit van die snitprobleem)

Laat 'n optimale oplossing vir die lineêre omsluiting van 'n $K/E/G$ -HTP gegee word deur $\underline{x} = (u_1, u_2, \dots, u_{r-1}, R, l_{r+1}, \dots, l_n)$, met $R \in (l_r, u_r)$. Dan sal \underline{x} ook 'n optimale oplossing vir die snitprobleem wees.

Die konsep van 'n snitprobleem laat die volgende idee ontstaan. In plaas daarvan om die ouer en twee kinders te vertak, soos wat Algoritme 2.1 (gewone vertak-en-begrens-algoritme) voorstel, kan dit eerder in vier subprobleme verdeel word. Indien funksie k die tob-funksie is, beteken dit dat funksie r verdeel word in 'n ondersnitkindprobleem waarin funksie r oor die interval $[u_r, v_r]$ strek, die snitprobleem waarin funksie r oor die interval $[v_r, w_r]$ strek en die bo-snitkindprobleem waarin funksie r se definisieversameling oor die interval $[w_r, u_r]$ strek. Uit Afleiding 5.1 is dit duidelik dat die optimale oplossing vir die snitprobleem reeds optimaal bepaal is (dit is dieselfde as die oplossing van die ouerprobleem), so daardie takke hoef nie verder ondersoek te word nie. Dus hoef daar steeds net in hoogstens twee subprobleme vertak te word, maar die vereniging van hierdie twee subprobleme (ondersnitkindprobleem en bo-snitkindprobleem) se toelaatbare gebiede is kleiner (of hoogstens gelyk) aan die toelaatbare gebied van die ouerprobleem. Die definisieversameling van funksie r word dus in hierdie geval nie net vertak nie, maar ook gesny (met die vereniging van die bo-snit en die ondersnit). Dit impliseer dat daar tydens elke vertakking potensieel 'n deel van die toelaatbare gebied afgesny kan word omdat die optimale oplossing vir daardie deel van die probleem reeds bepaal is. Op grond hiervan



(a) Die tob-funksie (f_r) van die ouerprobleem wat in die optimale oplossing van die lineêre omsluiting 'n toedeling van $x_r = R$ kry.

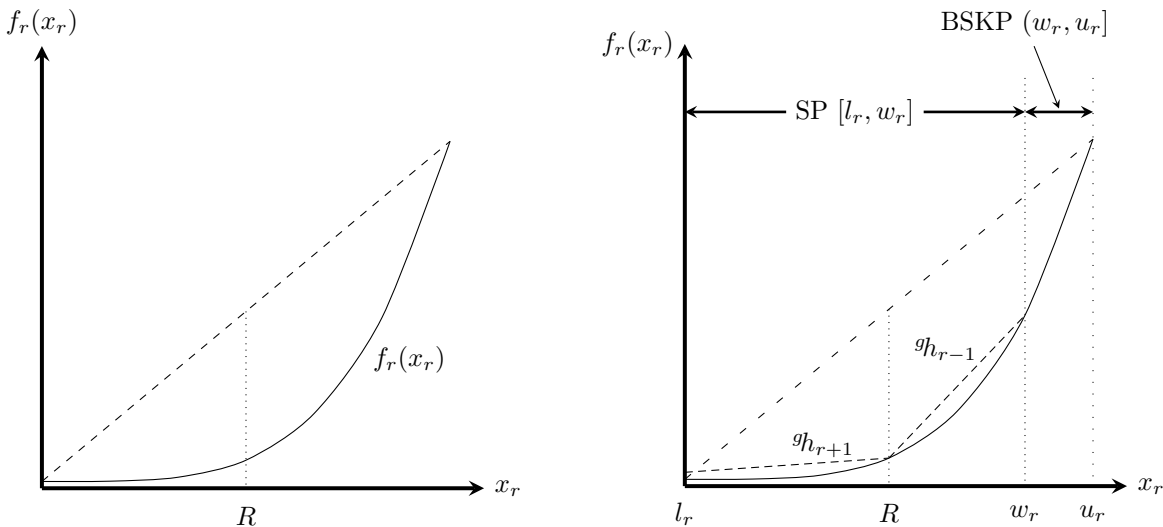
(b) Skep twee subprobleme, naamlik die OSKP en die BSKP, wat verder ondersoek moet word. Die derde subprobleem SP se optimale oplossing is reeds bekend en hoef dus nie verder ondersoek te word nie.

Figuur 5.6: Skematiese voorstelling van die geval waar $v_r \in (l_k, R)$ en $w_r \in (R, u_k)$.

kan verwag word dat die konvergensie van hierdie vertak-sny-en-begrensing vinniger sal geskied. Afhangende van die rykwydte van die snitprobleem se interval, dit is (v_r, w_r) , kan een van vier situasies ontstaan:

1. $v_r \in (l_k, R]$ en $w_r \in [R, u_k)$
Twee subprobleme, naamlik die ondersnitkind en die bo-snitkindprobleem word geskep. Sien Figuur 5.6(b) vir 'n skematiese voorstelling hiervan.
2. $v_r = l_r$ en $w_r \in [R, u_k)$
Slegs die bo-snitkindprobleem word geskep. Sien Figuur 5.7(b) vir 'n skematiese voorstelling hiervan.
3. $v_r \in (l_k, R]$ en $w_r = u_k$
Slegs die ondersnitkindprobleem word geskep. Sien Figuur 5.8(b) vir 'n skematiese voorstelling hiervan.
4. $v_r = l_k$ en $w_r = u_k$
Geen subprobleem word geskep nie. Sien Figuur 5.9(b) vir 'n skematiese voorstelling hiervan.

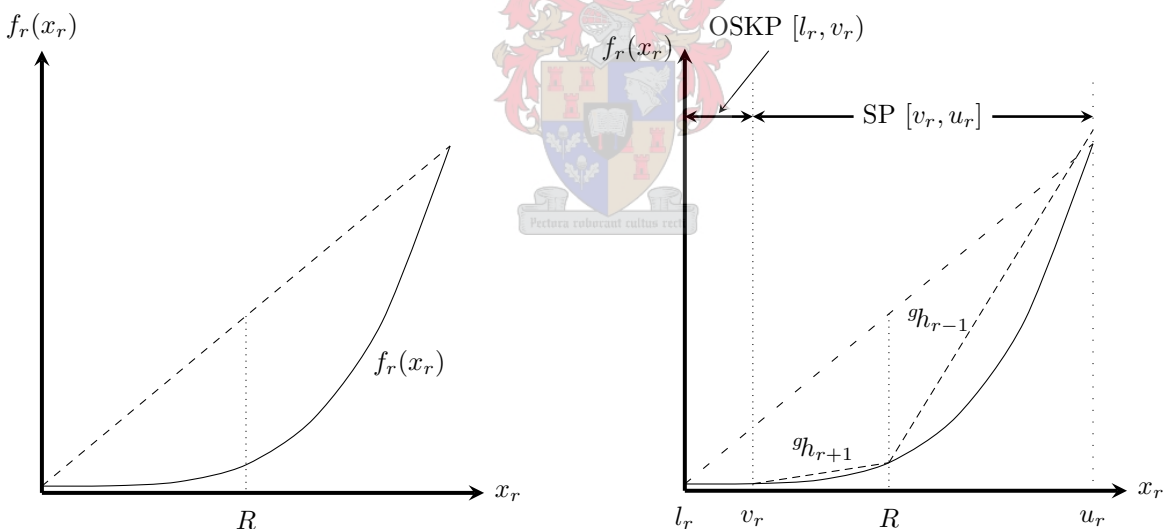
Die nodige terminologie is nou geskep om die werking van 'n algoritme te beskryf wat die ondersnitkindprobleem en 'n bo-snitkindprobleem gebruik om die vertak-en-begrensing vinniger te beëindig. Los die lineêre omsluiting van 'n *K/E/G*-HTP met behulp van Algoritme 2.4 op. Volgens die normale reëls vir vertak-en-begrensing behoort die tob-funksie van hierdie oplossing in twee subprobleme vertak te word, maar met bostaande



(a) Die tob-funksie (f_r) van die ouerprobleem wat in die optimale oplossing van die lineêre omsluiting 'n toedeling van $x_r = R$ kry.

(b) Skep een subprobleem, naamlik die BSKP, wat verder ondersoek moet word. Die ander subprobleem SP se optimale oplossing is reeds bekend en hoef dus nie verder ondersoek te word nie.

Figuur 5.7: Skematiese voorstelling van die geval waar $v_r = l_r$ en $w_r \in [R, u_k]$.



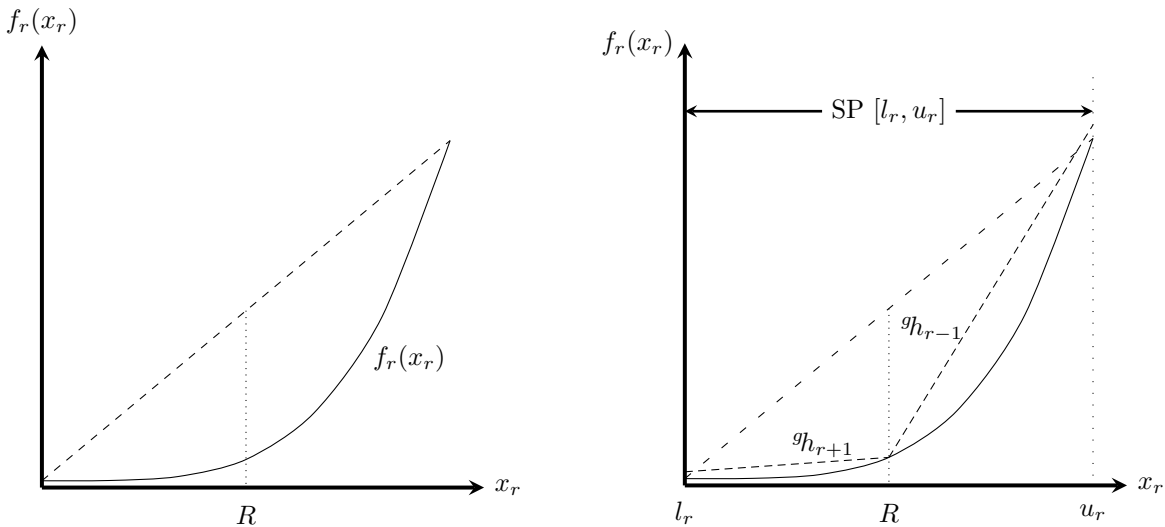
(a) Die tob-funksie (f_r) van die ouerprobleem wat in die optimale oplossing van die lineêre omsluiting 'n toedeling van $x_r = R$ kry.

(b) Skep een subprobleem, naamlik die OSKP wat verder ondersoek moet word. Die ander subprobleem SP se optimale oplossing is reeds bekend en hoef dus nie verder ondersoek te word nie.

Figuur 5.8: Skematiese voorstelling van die geval waar $v_r \in (l_k, R]$ en $w_r = u_k$.

kennis kan die probleem nou in nul, een of twee subprobleme vertak word. Hierdie metode om die probleem op te los, word in die onderstaande stappe beskryf.

1. [Inisialiseer en stel die beginprobleem op]



(a) Die tob-funksie (f_r) van die ouerprobleem wat in die optimale oplossing van die lineêre omsluiting 'n toedeling van $x_r = R$ kry.

(b) Skep geen subprobleme wat verder ondersoek moet word nie. Die enigste subprobleem is SP en sy optimale oplossing is reeds bekend en hoef dus nie verder ondersoek te word nie.

Figuur 5.9: Skematiese voorstelling van die geval waar $v_r = l_k$ en $v_r = u_k$.

Stel die totale aantal subprobleme, sowel as die indeks (t) daarvan gelyk aan 1 en skep die liniëring van die wortelprobleem. Stel die beste doelfunksiewaarde (begrenswaarde) gelyk aan nul.

2. [Los die volgende subprobleem op]
Los die t -de lineêre HTP, naamlik H_t op. Laat die optimale oplossing daarvoor \underline{x}^t wees.
3. [Doen die begrensing]
Toets of die lyndoelfunksiewaarde van H_t , naamlik $D_L(\underline{x}^t)$, groter/kleiner as die bestaande grens (G) is:
 - (a) [$D_L(\underline{x}^t) > G$] Bepaal die werklike doelfunksiewaarde van H_t , naamlik $D_w(\underline{x}^t)$. Indien hierdie waarde groter as die grens is, werk die grens by met hierdie waarde. Indien daar geen tob-funksie is nie, gaan na Stap 6. Indien daar 'n tob-funksie is, gaan na Stap 4. Gestel die tob-funksie is funksie r en die funksie kry 'n toedeling $x_r^t = R$, met $l_r^t < R < u_r^t$.
 - (b) [$D_L(\underline{x}^t) \leq G$] Gaan na Stap 6.
4. [Bepaal die ondersnit, dit wil sê bepaal die waarde van v_r , en bepaal OSKP indien $v_r > l_i^t$].
As $r = n$, dan moet die snit met die huidige helling bepaal word, oftewel $C = c_k$, anders moet die volgende funksie se helling gebruik word, dit wil sê $C = c_{k+1}$.
As $f_r(l_r^t) < f_r(R) + C(l_r^t - R)$, dan $v_r = l_r^t$ en gaan na Stap 5.
As $f_r'(R) < C$, dan $v_r = R$ en gaan na Stap 6.

Bepaal die snypunt v_r van $f_r(x)$ en $f_r(R) - C(R - x)$. Skep en voeg die OSKP by die versameling van subprobleme wat nog opgelos moet word.

5. [Bepaal die bo-snit, met ander woorde bepaal die waarde van w_r , en skep die BSKP indien $w_r < u_i^t$].

As $r = 1$, dan moet die snit met die huidige helling bepaal word, dit wil sê $C = c_r$, anders moet die vorige funksie se helling gebruik word, dit wil sê $C = c_{r-1}$.

As $f_r(u_r^t) < f_r(R) + C(u_k^t - R)$, dan $w_r = u_k^t$ en gaan na Stap 6.

As $f_r'(R) > C$, dan $w_r = R$ en gaan na Stap 6.

Bepaal die snypunt w_r van $f_r(x)$ en $f_r(R) + C(x - R)$. Skep en voeg die BSKP by die versameling van subprobleme wat nog opgelos moet word.

6. [Toets of al die subprobleme opgelos is]
Indien $t = T$, stop, anders laat $t = t + 1$. Gaan na Stap 2.

Vervolgens word die pseudokode van hierdie algoritme in Algoritme 5.4 gegee.

Algoritme 5.4 (Vertak-sny-en-begrens)

Doel: Bepaal 'n optimale oplossing vir 'n $K/E/G$ -HTP.

$T \leftarrow t \leftarrow 1$

$\underline{x}^b \leftarrow (0, 0, \dots, 0)$

$D^b \leftarrow 0$

vir $i \leftarrow 1$ **tot** n **doen**

$c_i^1 = \frac{f_i(u_i) - f_i(l_i)}{u_i - l_i}$

$u_i^1 \leftarrow u_i$

$l_i^1 \leftarrow l_i$

end vir



▷ Skep L^1

Stap 1

$\underline{x}^t \leftarrow \text{PSABG}(L^t)$

▷ Los L^t met behulp van Algoritme 2.10 op.

Stap 2

vir $i \leftarrow 1$ **tot** n **doen**

as $x_i^t > l_i^t$ **en** $x_i^t < u_i^t$ **dan**

$r \leftarrow i$

$R \leftarrow x_i^t$

ontsnap vir

anders

$r \leftarrow 0$

end as

end vir

as $D_L(\underline{x}^t) > D^b$ **dan**

as $D_w(\underline{x}^t) > D^b$ **dan**

$D^b \leftarrow D_w(\underline{x}^t)$

▷ Bepaal die tob-funksie

▷ Doen die begrensing

```

 $\underline{x}^b \leftarrow \underline{x}^t$ 
as  $r > 0$  dan
    Gaan na Stap 3
anders
    Gaan na Stap 5
end as
end as
anders
    Gaan na Stap 5
end as

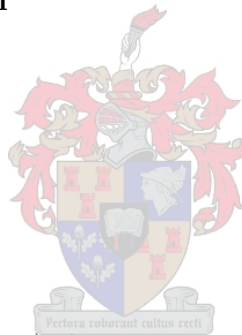
```

Stap 3

```

as  $r = n$  dan
     $C \leftarrow c_n^t$ 
anders
     $C \leftarrow c_{r+1}^t$ 
end as
as  $f_r(l_r^t) < f_r(R) + C(l_r^t - R)$  dan
    Gaan na Stap 5
anders as  $f_r'(R) < C$  dan
     $v_r \leftarrow R$ 
anders
     $A = l_r^t$ 
     $B = R$ 
    herhaal
         $v_r = \frac{B-A}{2}$ 
        as  $f_r(v_r) < f_r(R) - C(R - v_r)$  dan
             $B = v_r$ 
        anders
             $A = v_r$ 
        end as
    totdat  $0 \leq f_r(R) - C(R - v_r) - f_r(v_r) \leq \delta$ 
end as
 $T = T + 1$ 
vir  $i \leftarrow 1$  tot  $n$  doen
     $l_i^T \leftarrow l_i^t$ 
     $u_i^T \leftarrow u_i^t$ 
     $c_i^T \leftarrow \frac{f_i(u_i^t) - f_i(l_i^t)}{u_i^t - l_i^t}$ 
end vir
 $u_r^t \leftarrow v_r$ 
 $c_r^T \leftarrow \frac{f_r(u_r^t) - f_r(l_r^t)}{u_r^t - l_r^t}$ 

```



$\triangleright \delta$ is 'n gekose akkuraatheid

Stap 4

```

as  $r = 1$  dan

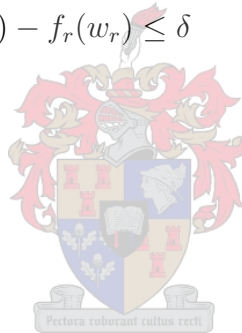
```

```

     $C \leftarrow c_1^t$ 
anders
     $C \leftarrow c_{r-1}^t$ 
end as
as  $f_r(u_r^t) < f_r(R) + C(w_r - R)$  dan
    Gaan na Stap 5
anders as  $f_r'(R) > C$  dan
     $w_r \leftarrow R$ 
anders
     $A = R$ 
     $B = u_r^t$ 
herhaal
     $w_r = \frac{B-A}{2}$ 
as  $f_r(w_r) < f_r(R) + C(w_r - R)$  dan
     $A = w_r$ 
anders
     $B = w_r$ 
end as
totdat  $0 \leq f_r(R) + C(w_r - R) - f_r(w_r) \leq \delta$ 
end as
 $T = T + 1$ 
vir  $i \leftarrow 1$  tot  $n$  doen
     $l_i^T \leftarrow l_i^t$ 
     $u_i^T \leftarrow u_i^t$ 
     $c_i^T \leftarrow \frac{f_i(u_i^t) - f_i(l_i^t)}{u_i^t - l_i^t}$ 
end vir
 $l_r^T \leftarrow w_r$ 
 $c_r^T \leftarrow \frac{f_r(u_r^T) - f_r(l_r^T)}{u_r^T - l_r^T}$ 

```

$\triangleright \delta$ is 'n gekose akkuraatheid



Stap 5

```

as  $t = T$  dan
    Stop
anders
     $t = t + 1$ 
    Gaan na Stap 1
end as

```

5.5 Isogrens-vertak-sny-en-begrens-algoritme

In die isogrens-vertak-sny-en-begrens-algoritme word die tegniek waarop die isogrens-vertak-en-begrens-algoritme sowel as die tegniek waarop die vertak-sny-en-begrens-algoritme berus, gekombineer.

'n Beskrywing van hoe die stappe van die isogrens-vertak-sny-en-begrens-algoritme werk, volg hieronder.

1. [Inisieer]
Stel al die indekse en veranderlikes se waardes vas en skep die eerste probleem in \mathcal{I}^o om op te los.
2. [Maak 'n siklus wat al die subprobleme in \mathcal{I}^o en \mathcal{I}^b oplos]
 - (a) [Los subprobleem p op met behulp van ondergrensprosedure]
Roep die iso-ondergrensprosedure om subprobleem $p \in \mathcal{I}^o$ op te los.
 - (b) [Los subprobleem d op met behulp van bogrensprosedure]
Roep die iso-bogrensprosedure om subprobleem $d \in \mathcal{I}^b$ op te los.
3. [Uitset: Optimale oplossing]
Die optimale doelfunksiewaarde is D^b met ooreenstemmende oplossing \underline{x}^b soos wat in beide die ondergrens- en bogrensalgoritmes bygewerk is. Stop.

Die pseudokode vir hierdie algoritme word in Algoritme 5.5 gegee. Die voetskrif [i] beteken dat daar altyd met die funksies, veranderlikes, ensovoorts gewerk word wat volgens die funksies se gemiddelde helling van die grootste tot die kleinste gesorteer is.

Algoritme 5.5 (Isogrens)

Prosedure ISOGRENS(HTP)

$Kl_d \leftarrow 0$

$Kl_p \leftarrow 0$

$P \leftarrow 1$

$D \leftarrow 0$

$p \leftarrow 0$

$d \leftarrow 0$

$T \leftarrow 0$

herhaal

terwyl $p < P$ **doen**

$p \leftarrow p + 1$

as $D_L(\underline{x}^p) > D_m$ **dan**

Roep: ONDERGRENS (L_p^o)

ontsnap **terwyl**

end as

end **terwyl**

terwyl $d < D$ **doen**

$d \leftarrow d + 1$

as $D_L(\underline{x}^d) > D_m$ **dan**

Roep: BOGRENS (L_d^b)

ontsnap **terwyl**




```

    end as
  end terwyl
  as  $p \geq P$  dan
     $Kl_p \leftarrow 1$ 
  anders
     $Kl_p \leftarrow 0$ 
  end as
  as  $d \geq D$  dan
     $Kl_d \leftarrow 1$ 
  anders
     $Kl_d \leftarrow 0$ 
  end as
  totdat  $Kl_d = 1$  en  $Kl_p = 1$ 
end Prosedure
Prosedure ONDERGRENS(HTP)
  vir  $i \leftarrow 1$  tot  $n$  doen
     $x_i^t \leftarrow l_i^p$ 
     $B_t \leftarrow B_t - l_i^p$ 
     $S_T \leftarrow S_T + u_i^p$ 
  end vir
  herhaal
    vir  $i \leftarrow 1$  tot  $n$  doen
      as  $u_{[i]}^{o,p} - l_{[i]}^{o,p} \leq B_t$  dan
         $x_{[i]}^{o,p} \leftarrow u_{[i]}^{o,p}$ 
         $B_t \leftarrow B_t - u_{[i]}^{o,p} + l_{[i]}^{o,p}$ 
         $T_s \leftarrow -1$ 
      anders
         $x_{[i]}^{o,p} \leftarrow l_{[i]}^{o,p} + B_t$ 
         $T_s \leftarrow 0$ 
         $r \leftarrow i$ 
      ontsnap vir
    end as
  end vir
  as  $T_s = -1$  en  $B_t > 0$  dan
    ontsnap prosedure
  end as
  as  $D_L \underline{x}^{o,p} < D^b$  dan
    ontsnap herhaal
  end as
  as  $D^b < D_w(\underline{x}^{o,p})$  dan
     $D^b \leftarrow D_w(\underline{x}^{o,p})$ 
     $\underline{x}^b \leftarrow \underline{x}^{o,p}$ 
  end as
  as  $T_s = -1$  dan

```



▷ Los die subprobleem op

▷ As $D_L(\underline{x})$ swakker as D^b , stop die vertakking

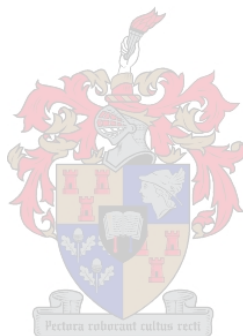
▷ Werk D^b by

▷ Stop indien daar soveel afgesny is dat B_t te veel is

```

    ontsnap herhaal
  end as
  as  $B_t = 0$  dan
    ontsnap herhaal
  anders
    bepaal  $w_r$ 
    as  $w_r < u_r^{o,p}$  dan
       $D \leftarrow D + 1$ 
       $H_D^b \leftarrow H_p^o$ 
       $l_r^{b,D} \leftarrow w_r$ 
       $\mathcal{I}^b \leftarrow \mathcal{I}^b \cup H_D^b$ 
    end as
    bepaal  $v_r$ 
    as  $v_r > l_r^{o,p}$  dan
       $u_r^{o,p} \leftarrow v_r$ 
    anders
      ontsnap prosedure
    end as
  end as
  totdat
end Prosedure
Prosedure BOGRENS(HTP)
  vir  $i \leftarrow 1$  tot  $n$  doen
     $x_i^t \leftarrow u_i^d$ 
     $B_t \leftarrow B_t - l_i^d$ 
     $S_{og} \leftarrow S_{og} + l_i^d$ 
     $S_{bg} \leftarrow S_{bg} + u_i^d$ 
  end vir
   $B_t \leftarrow S_{bg} - B$ 
  as  $B_t < 0$  of  $S_{og} > B$  dan
    ontsnap prosedure
  end as
  herhaal
    vir  $i \leftarrow n$  tot 1 doen Stap -1
      as  $u_{[i]}^{b,d} - l_{[i]}^{b,d} \leq B_t$  dan
         $x_{[i]}^{b,d} \leftarrow l_{[i]}^{b,d}$ 
         $B_t \leftarrow B_t - u_{[i]}^{b,d} + l_{[i]}^{b,d}$ 
         $T_s \leftarrow -1$ 
      anders
         $x_{[i]}^{b,d} \leftarrow u_{[i]}^{b,d} - B_t$ 
         $T_s \leftarrow 0$ 
         $r \leftarrow i$ 
      ontsnap vir
    end as

```



▷ Stop as daar geen tob-funksie is nie

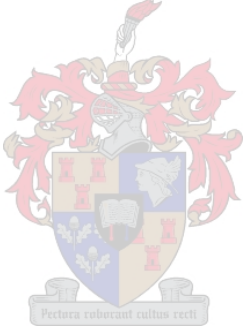
▷ Los die subprobleem op

```

end vir
as  $T_s = -1$  en  $B_t > 0$  dan
  ontsnap vir
end as
as  $D_L \underline{x}^{b,d} < D^b$  dan
  ontsnap herhaal
end as
as  $D^b < D_w(\underline{x}^{o,p})$  dan
   $D^b \leftarrow D_w(\underline{x}^{o,p})$ 
   $\underline{x}^b \leftarrow \underline{x}^{o,p}$ 
end as
as  $T_s = -1$  dan
  ontsnap herhaal
end as
as  $B_t = 0$  dan
  ontsnap herhaal
anders
  bepaal  $v_r$ 
  as  $v_r > l_{[r]}^{b,d}$  dan
     $P \leftarrow P + 1$ 
     $H_P^o \leftarrow H_d^b$ 
     $u_{[r]}^{o,P} \leftarrow v_r$ 
     $\mathcal{I}^o \leftarrow \mathcal{I}^o \cup H_P^o$ 
  end as
  bepaal  $w_r$ 
  as  $w_r < u_{[r]}^{b,d}$  dan
     $l_{[r]}^{b,d} \leftarrow w_r$ 
  anders
    ontsnap prosedure
  end as
end as
totdat  $D_L(\underline{x}^{b,d}) < D^b$ 
end Prosedure

```

▷ As $D_L(\underline{x})$ swakker as D^b , stop die vertakking
 ▷ Werk D^b by
 ▷ Stop indien daar soveel afgesny is dat B_t te veel is
 ▷ Stop as daar geen tob-funksie is nie
 ▷ Skep 'n subprobleem een plaas by \mathcal{I}^o



5.5.1 Numeriese metode om w_r te bepaal

Die prosedure hieronder kan ingespan word om w_r , of met 'n geringe aanpassing, v_r te bepaal.

```

as  $r = 1$  dan
   $M \leftarrow \mathcal{H}_i$ 
anders
   $M \leftarrow \mathcal{H}_{i-1}$ 
end as

```

```

as  $f_r(l_r) + M(R - l_r) - f_r(u_r) \geq 0$  dan
     $w_r \leftarrow u_r$ 
    ontsnap procedure
end as
as  $f'_r R \geq M$  dan
     $w_r \leftarrow R$ 
    ontsnap procedure
end as
 $A \leftarrow R$ 
 $B \leftarrow u_r$ 
herhaal
     $W \leftarrow (A + B)/2$ 
     $f_{out} \leftarrow f_r(A) + M(W - A) - f_r(W)$ 
    as  $f_{out} > 0$  dan
         $A \leftarrow W$ 
    anders
         $B \leftarrow W$ 
    end as
totdat  $f_{out} \geq 0$  en  $f_{out} < \delta$ 
 $w_r \leftarrow W$ 

```

In die prosedure hierbo is δ 'n gekose akkuraatheid waarmee w_r bereken moet word.

5.6 Rangorde en aantal eksperimente



Een van die eerste vrae wat ontstaan, is hoeveel eksperimente nodig is om te kan beweer dat een algoritme 'n beduidende beter oplossingstyd as 'n ander het. Een so 'n manier is die metode van Dudewicz en Dalal [11] om verskillende simulaties met mekaar te vergelyk en 'n rangorde op te stel om te bepaal watter een die vinnigste is.

Hierdie metode het twee aannames, naamlik dat die oplossingstye normaal verdeel moet wees en dat die oplossingstye onafhanklik van mekaar moet wees. In hierdie geval word aan altwee hierdie aannames voldoen.

Dudewicz en Dalal se algoritme bestaan uit die volgende agt stappe:

1. Doen $n_0 > 2$ aanvanklike eksperimente vir elke metode. Dudewicz en Dalal beveel egter aan, soos wat hulle tabel ook aandui, dat 20 of 40 aanvanklike eksperimente gedoen word.
2. Bereken die gemiddelde $\bar{X}_i^{(1)}$ en variansie s_i^2 vir die n_0 waarnemings van elk van die k verskillende metodes.
3. Bereken die totale aantal eksperimente N_i wat vir elke metode nodig is met behulp

van

$$N_i = \text{maks} \left(n_0 + 1, \left\lceil \frac{h_1^2 s_i^2}{\delta^{*2}} \right\rceil \right), \quad (5.3)$$

waar h_1 Dudewicz en Dalal se konstante, s_i die variansie van die eerste n_0 eksperimente, δ^* 'n gekose grootheid wat aandui hoeveel die een algoritme beter as die ander moet wees voordat dit beter genoeg kan word en $\lceil x \rceil$ die kleinste heelgetal is wat groter as x is.

4. Doen $N_i - n_0$ eksperimente met elke metode.
5. Bepaal die gemiddelde $\bar{X}_i^{(2)}$ van die $N_i - n_0$ eksperimente vir elke metode.
6. Bepaal die gewigte

$$W_{i1} = \frac{n_0}{N_i} \left[1 + \sqrt{1 - \frac{n_0}{N_i} \left(1 - \frac{(N_i - n_0)\delta^{*2}}{h_1^2 s_i^2} \right)} \right] \quad \text{en} \quad (5.4)$$

$$W_{i2} = 1 - W_{i1} \quad (5.5)$$

vir elke metode. Al die veranderlikes het die betekenis soos in die voorafgaande stappe.

7. Bepaal die geweege gemiddelde

$$\bar{X}_i = W_{i1} \bar{X}_i^{(1)} + W_{i2} \bar{X}_i^{(2)}. \quad (5.6)$$

8. Die k metodes kan met P^* sekerheid gerangskik word in die volgorde van \bar{X}_i , waar elke metode δ^* beter as die ander is.

P^*	n_0	k			
		2	3	4	5
90%	20	1.896	2.342	2.583	2.747
90%	40	1.852	2.283	2.514	2.669
95%	20	2.453	2.872	3.101	3.258
95%	40	2.386	2.786	3.003	3.150

Tabel 5-1: Tabel vir die bepaling van h_1 , Dudewicz en Dalal [11] se konstante. In hierdie tabel is P^* die persentasie sekerheid waarmee die een eksperiment beter as die ander moet wees, n_0 die aanvanklike aantal eksperimente en k die aantal metodes wat teen mekaar gemeet en gerangskik moet word.

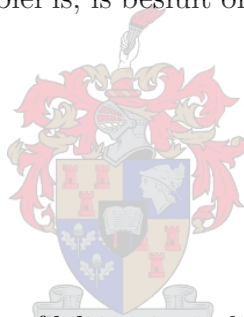
Indien die bostaande agt stappe uitgevoer word, word die resultate in Tabel 5-2 verkry. Kies $n_0 = 20$, $\delta^* = 0.01$ en $P^* = 95\%$. Die aantal metodes, k , is 4 en $h_1 = 3.101$, afgelees uit Tabel 5-1.

$N_i - n_0$ kan ook gebruik word as 'n aanduiding van hoeveel eksperimente (herhalings) nodig is om beduidende resultate te genereer. Uit die bostaande berekeninge is dit duidelik dat daar vir die drie nuwe algoritmes min (slegs een!) herhalings nodig is, terwyl

	VSB	IGVSB	VB	IGVB
$\bar{X}_i^{(1)}$	0.031	0.028	0.430	0.039
s_i	0.010	0.007	1.096	0.009
N_i	21	21	115533	21
$\bar{X}_i^{(2)}$	0.031	0.027	1.028	0.047
W_{i1}	1.177	1.372	0.000	1.253
W_{i2}	-0.177	-0.372	1.000	-0.253
\bar{X}_i	0.031	0.029	1.028	0.037
Rangorde	2	1	4	3

Tabel 5-2: Rangorde vir die vier verskillende tipes vertak-en-begrens-algoritmes. Die volgende afkortings word gebruik: VSB = vertak-sny-en-begrens-algoritme, IGVSB = isogrens-vertak-sny-en-begrens-algoritme, VB = vertak-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.

heelwat meer (ongeveer 115 000) vir die gewone vertak-en-begrens-algoritme nodig is. Aangesien dit te lank sal neem om elke keer 115 000 herhalings te doen en omdat die gemiddelde oplossingstye relatief stabiel is, is besluit om alle resultate weer te gee vir 100 herhalings.



5.7 Samevatting

Die samevatting bestaan uit drie onderafdelings, naamlik 'n onderafdeling oor die metodologie, 'n onderafdeling met resultate en 'n onderafdeling waarin gevolgtrekkings verskaf word oor die resultate.

5.7.1 Metodologie

Die metodologie is soortgelyk aan dié in die vorige hoofstukke. In die afwesigheid van 'n standaard stel probleme word probleme lukraak genereer en opgelos. Dieselfde 100 funksies is vir elke keuse van n met al vier die algoritmes opgelos. Die metodologie is soortgelyk aan die metodologie wat in §3.6.1 vir vaste intervale en §4.3.2 vir wisselende intervale verskaf word. Die enkele gevalle waar waar daar wel afgewyk word van hierdie metodologie, word die afwyking in die betrokke byskrif van die tabel of grafiek verskaf.

Die probleme word met dieselfde rekenaar as in §3.6.1 opgelos.

Die kode vir die algoritmes wat in hierdie hoofstuk aangebied word, kan op die CD agterin hierdie proefskrif in die lêer met lêernaam `keg.xls` gevind word.

5.7.2 Resultate

Die resultate vir prestasie van die verskillende algoritmes vir die oplossing van 'n $K/E/G$ -HTP word in Tabel 5-3 gegee.

n	VSB	IGVSB	VB	IGVB
5	0.258	0.141	0.281	0.070
10	0.188	0.240	0.267	0.388
15	0.633	0.539	1.008	0.328
20	0.932	0.727	4.318	0.814
25	1.313	0.867	15.943	0.879
30	1.998	1.277	33.041	1.283
35	1.811	1.201	52.043	1.717
40	2.801	1.811	106.787	2.191
45	3.510	2.426	171.738	2.777
50	4.254	2.672	238.242	3.105
55	4.178	2.813	217.447	2.871
60	5.619	3.422	292.102	4.148

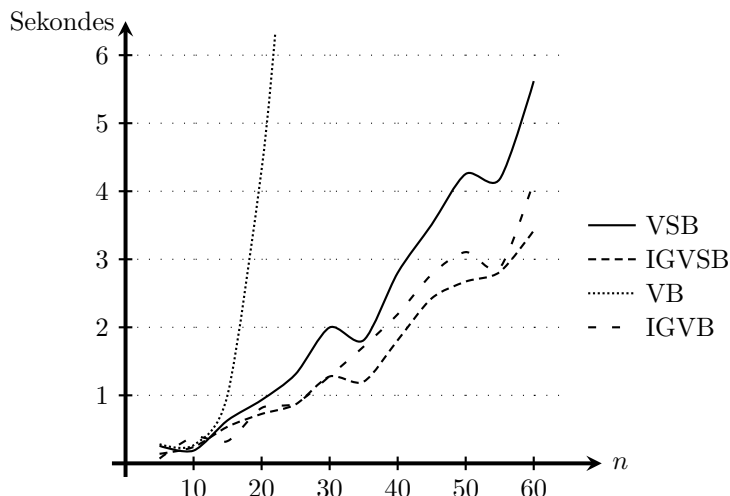
Tabel 5-3: Die totale oplossingstyd in sekondes van 100 lukraak gegenereerde $K/E/G$ -HTP's teenoor die aantal funksies n . Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die volgende afkortings word gebruik: VSB = vertak-sny-en-begrens-algoritme, IGVSB = isogrens-vertak-sny-en-begrens-algoritme, VB = vertak-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.

'n Grafiese voorstelling van hierdie data word in Figuur 5.10 verskaf. Uit die data kan die gevolgtrekking gemaak word dat die verbeterde vertak-en-begrens-algoritmes vir hierdie keuse van probleme beduidend beter vaar as die gewone vertak-en-begrens-algoritme. Meer spesifiek gebruik al drie die verbeterde algoritmes ongeveer 1 - 1.5% van die tyd om dieselfde probleem op te los as die gewone-vertak-en-begrens algoritmes. Dit impliseer 'n byna 99% tydsbesparing. 'n Ander manier om daarna te kyk, is dat die nuwe algoritmes dieselfde probleme ongeveer 100 maal vinniger oplos, wat beslis 'n groot verbetering op die bestaande vertak-en-begrens-algoritme is.

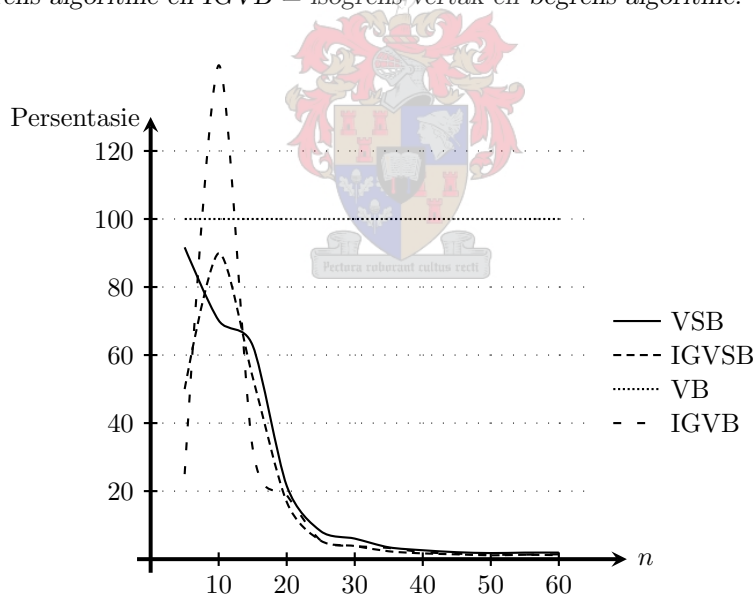
'n Ander manier om die data visueel voor te stel is om die oplossingstye van die onderskeie algoritmes uit te druk relatief tot die tyd wat die gewone vertak-en-begrens-algoritme neem om dieselfde probleme op te los. Hierdie resultate word in Figuur 5.11 gegee.

Die grafieke van die aantal subprobleme wat elke algoritme skep om dieselfde probleme op te los, word in Figuur 5.12 gegee en het min of meer dieselfde patroon as die soortgelyke grafiek van die oplossingstye vir elke algoritme. Analooq aan Figuur 5.11 word die persentasie van die aantal subprobleme wat elke algoritme skep relatief tot dié van die gewone vertak-en-begrens-algoritme in Figuur 5.13 gegee.

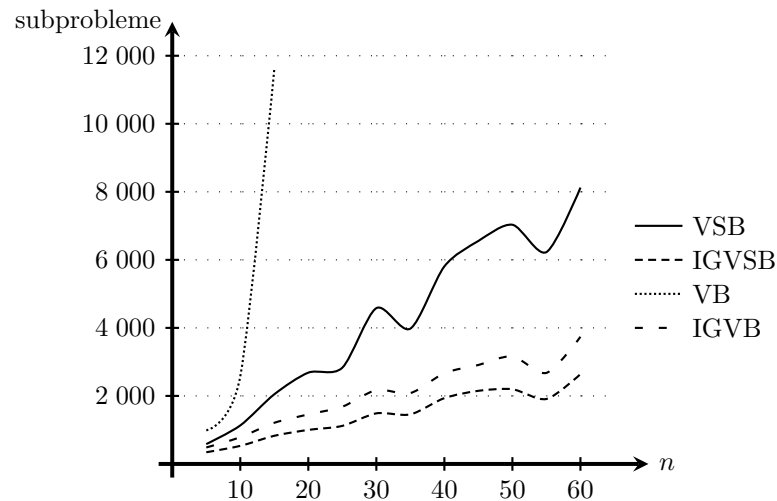
Alhoewel die nuwe algoritmes wat hier voorgestel word, minder subprobleme skep as gewone vertak en begrens, neem dit hierdie algoritmes langer om 'n enkele subprobleem op te los. Die relatiewe oplossingstye per subprobleem vir elk van die algoritmes word in



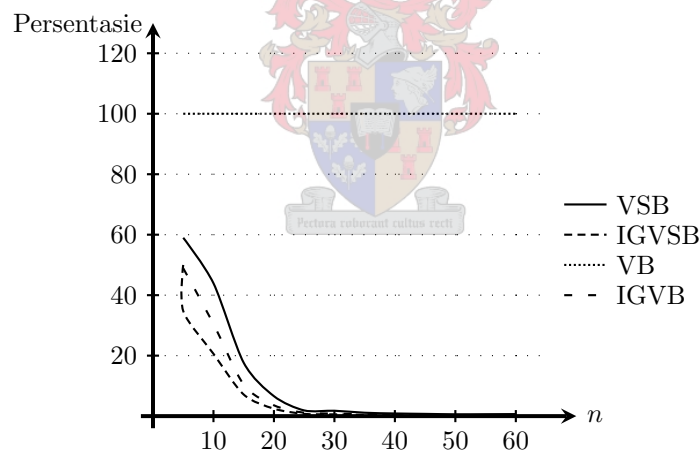
Figuur 5.10: Grafieke van die totale oplossingstyd in sekondes van 100 lukraak gegeneerde $K/E/G$ -HTP's teenoor die aantal funksies n . Die veranderlikes se bo-grense is lukraak gegeneer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegeneer. Die volgende afkortings word gebruik: VSB = vertak-sny-en-begrens-algoritme, IGVSb = isogrens-vertak-sny-en-begrens-algoritme, VB = vertak-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.



Figuur 5.11: Grafieke van die persentasie van die oplossingstyd relatief tot die oplossings-tyd van die normale vertak-en-begrens-algoritme om 100 lukraak gegeneerde $K/E/G$ -HTP's op te los teenoor die aantal funksies n . Die veranderlikes se bo-grense is lukraak gegeneer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegeneer. Die volgende afkortings word gebruik: VSB = vertak-sny-en-begrens-algoritme, IGVSb = isogrens-vertak-sny-en-begrens-algoritme, VB = vertak-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.



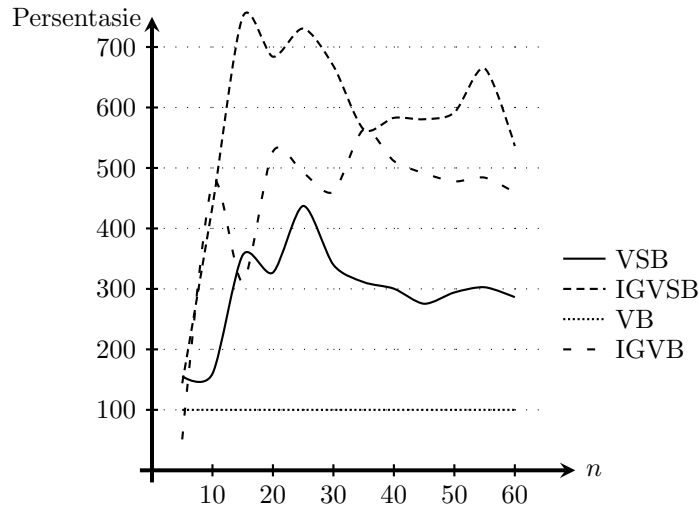
Figuur 5.12: Grafieke van die aantal subprobleme wat opgelos moet word om 100 lukraak gegenereerde $K/E/G$ -HTP's op te los teenoor die aantal funksies n . Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die volgende afkortings word gebruik: VSB = vertak-sny-en-begrens-algoritme, IGVSb = isogrens-vertak-sny-en-begrens-algoritme, VB = vertak-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.



Figuur 5.13: Grafieke van die persentasie van die aantal subprobleme relatief tot die aantal subprobleme van die normale vertak-en-begrens-algoritme om 100 lukraak gegenereerde $K/E/G$ -HTP's op te los teenoor die aantal funksies n . Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die volgende afkortings word gebruik: VSB = vertak-sny-en-begrens-algoritme, IGVSb = isogrens-vertak-sny-en-begrens-algoritme, VB = vertak-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.

5.7.2.1 Invloed van gemiddelde helling

In hierdie afdeling word die invloed van die intervallengte waaruit die gemiddelde hellings lukraak gegenereer word, ondersoek. Laat Ω die intervallengte wees waaruit die gemid-

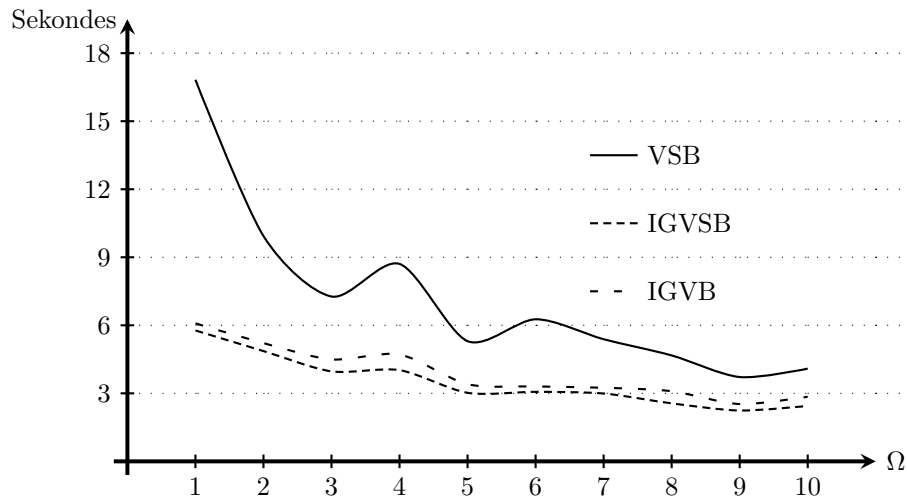


Figuur 5.14: Grafieke van die persentasie van die oplossingstyd in sekondes relatief tot die oplossingstyd van die normale vertak-en-begrens-algoritme per subprobleem van lukraak gegenereerde $K/E/G$ -HTP's op te los teenoor die aantal funksies n . Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die volgende afkortings word gebruik: VSB = vertak-sny-en-begrens-algoritme, IGVS = isogrens-vertak-sny-en-begrens-algoritme, VB = vertak-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.

delde hellings gekies word. Dit wil sê, indien die gemiddelde hellings lukraak uit die interval $[a, b]$ gegenereer word, sal $\Omega = b - a$. Dit is te wagte dat die aantal subprobleme wat tydens die vertak-en-begrens-algoritmes gegenereer word, en dus die oplossingstyd,

Ω	VSB	IGVS	VB	IGVB
1	16.828	5.773	504.039	6.086
2	9.938	4.863	457.172	5.215
3	7.270	3.969	417.840	4.496
4	8.707	4.023	431.801	4.711
5	5.301	3.023	386.348	3.402
6	6.266	3.066	361.527	3.305
7	5.387	2.992	327.289	3.246
8	4.672	2.559	278.176	3.094
9	3.719	2.242	281.711	2.527
10	4.090	2.445	271.207	2.852
20	3.559	2.277	187.938	2.457

Tabel 5-4: Die totale oplossingstyd om 100 lukraak gegenereerde $K/E/G$ -HTP's wat elk uit 60 funksies bestaan, op te los teenoor intervallengte Ω waaruit die gemiddelde hellings gegenereer is. Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die volgende afkortings word gebruik: VSB = vertak-sny-en-begrens-algoritme, IGVS = isogrens-vertak-sny-en-begrens-algoritme, VB = vertak-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.



Figuur 5.15: Grafieke van die totale oplossingstyd om 100 lukraak gegenereerde $K/E/G$ -HTP's wat elk uit 60 funksies bestaan, op te los teenoor intervallengte Ω waaruit die gemiddelde hellings gegenereer is. Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die volgende afkortings word gebruik: VSB = vertak-sny-en-begrens-algoritme, IGVS = isogrens-vertak-sny-en-begrens-algoritme, VB = vertak-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.

meer sal wees mits die gemiddelde hellings van die funksies nader aan mekaar is. 'n Eksperiment is gedoen deur lukraak probleme te genereer waarin die interval waaroor die gemiddelde hellings kan varieer, verstel kan word. In Tabel 5-4 word hierdie resultate getoon. Dieselfde resultate, sonder die data vir die gewone vertak-en-begrens-algoritme, volg in Figuur 5.15 omdat hierdie waardes ordes (ongeveer 100 maal) groter is as dié van die ander metodes. Geen resultate kon gegenereer word vir $\Omega < 1$ nie, omdat die aantal subprobleme wat die algoritmes skep in die oplossing van die probleem so toeneem dat dit nie in die toegelate 64 KB wat in VBA per subroetine gestoor kan word nie.

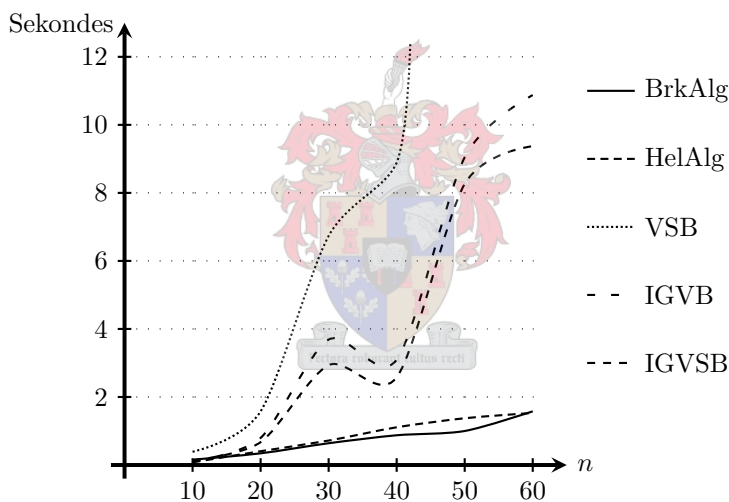
Uit Tabel 5-4 en Figuur 5.15 is dit duidelik dat die oplossingstyd vir al die algoritmes drasties afneem met 'n toename in die intervallengte Ω waaruit die gemiddelde helling gegenereer word. Die algoritmes se oplossingstye daal na tussen $\pm 21\%$ en $\pm 40\%$ van die oorspronklike oplossingstye. Die oplossingstyd vir die vertak-sny-en-begrens-algoritme vertoon die grootste verbetering, terwyl die isogrens-vertak-en-begrens-algoritme die kleinste verbetering toon.

5.7.2.2 Prestasie met die oplossing van $K/E/V$ -HTP

Die algoritmes wat in hierdie hoofstuk voorgestel word, kan natuurlik ook gebruik word om $K/E/V$ -HTP's op te los. 'n Eksperiment wat die oplossingstye van die algoritmes in Hoofstuk 3 en die algoritmes in hierdie hoofstuk met mekaar vergelyk, is gedoen. Vir hierdie eksperiment is die gemiddelde hellings gegenereer uit die interval $[5, 8]$. Die resultate word in Tabel 5-5 en Figuur 5.16 getoon.

n	BrkAlg	HelAlg	VSB	IGVSB	IGVB
10	0.1563	0.0781	0.3906	0.1094	0.1406
20	0.3438	0.4063	1.5781	0.7969	0.6719
30	0.6406	0.7188	6.7344	3.6875	2.9375
40	0.8750	1.1094	8.8906	3.0938	2.5781
50	1.0001	1.3750	26.4219	9.0469	8.2656
60	1.5781	1.5313	32.5313	10.8750	9.3750

Tabel 5-5: Die totale oplossingstyd in sekondes van 100 lukraak gegeneerde $K/E/V$ -HTP's teenoor die aantal funksies in die doelfunksie n . Die veranderlikes se bogrense is 200 eenhede. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegeneer. Die volgende afkortings word gebruik: BrkAlg = breukalgoritme, HelAlg = hellingalgoritme, VSB = vertak-sny-en-begrens-algoritme, IGVSB = isogrens-vertak-sny-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.



Figuur 5.16: Grafiek van die totale oplossingstyd in sekondes van 100 lukraak gegeneerde $K/E/V$ -HTP's teenoor die aantal funksies in die doelfunksie n . Die veranderlikes se bogrense is 200 eenhede. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegeneer. Die volgende afkortings word gebruik: BrkAlg = breukalgoritme, HelAlg = hellingalgoritme, VSB = vertak-sny-en-begrens-algoritme, IGVSB = isogrens-vertak-sny-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.

In Tabel 5-5 en Figuur 5.16 kan gesien word dat die algoritmes vir die spesiale geval van $K/E/V$ -HTP's wat in Hoofstuk 3 gegee word, die nuwe vertak-en-begrens-algoritmes wat in hierdie hoofstuk gegee word, uitpresteer. Beide die breuk- en die hellingalgoritme se oplossingstye is ongeveer 'n vyfde van die van dié beste vertak-en-begrens-algoritme indien die $K/E/V$ -HTP's met 60 funksies in die doelfunksie beskou word.

5.7.2.3 Prestasie van vertak-en-begrens-algoritmes vs isogrensheuristiek vir $K/E/G$ -HTP's

Die gemiddelde persentasie fout e_r in die doelfunksiewaardes van die oplossing vir die isogrensheuristiek word in Tabel 5-6 getoon. Die gemiddelde absolute fout e_a na m herhalings word gegee deur

$$e_a = \frac{1}{m} \sum_{i=1}^m D_w(i) - D_h(i)$$

en die gemiddelde persentasie fout e_r na m herhalings is bereken deur

$$e_r = \frac{100}{m} \sum_{i=1}^m \frac{D_w(i) - D_h(i)}{D_w(i)},$$

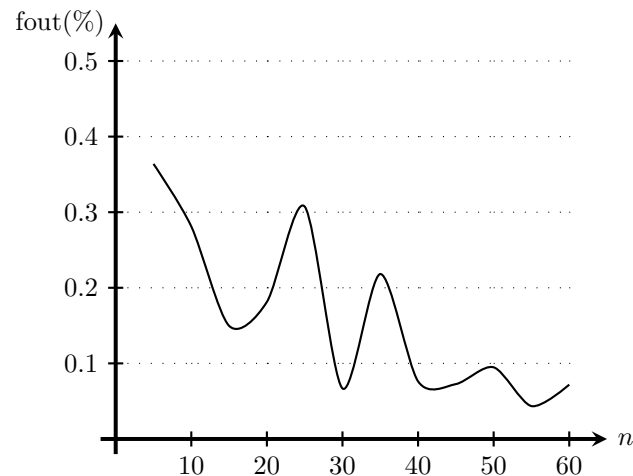
waar $D_w(i)$ die werklike optimale doelfunksiewaarde van probleem i is en $D_h(i)$ die doelfunksiewaarde vir probleem i is wat met behulp van die isogrensheuristiek verkry is. 'n Grafiek van hierdie data word in Figuur 5.17 getoon.

n	fout	fout (%)
5	8.7620	0.3639
10	10.3694	0.2811
15	6.1511	0.1497
20	11.7706	0.1813
25	16.2807	0.3074
30	8.0956	0.0668
35	12.7000	0.2182
40	9.6311	0.0762
45	9.7111	0.0726
50	9.6561	0.0948
55	8.1869	0.0435
60	7.6863	0.0719

Tabel 5-6: Gemiddelde persentasie fout van die isogrensheuristiek relatief tot die optimale doelfunksiewaarde vir 100 lukraak gegenereerde $K/E/G$ -HTP's teenoor verskillende hoeveelhede funksies in die doelfunksie n . Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is.

Uit die data wat in Tabel 5-6 gegee word, is dit duidelik dat die absolute fout van die heuristiek ongeveer konstant bly met 'n toename in die aantal funksies in die doelfunksie. Indien daar egter na die relatiewe fout gekyk word, neem dit af met 'n toename in die aantal funksies. Die relatiewe fout wissel van $\pm 0.3\%$ tot $\pm 0.07\%$, wat 'n goeie prestasie vir die isogrensheuristiek impliseer.

Die prestasie in terme van die oplossingstyd van die isogrensheuristiek versus die eksakte metodes vir die oplossing van $K/E/G$ -HTP's word in Tabel 5-7 getoon.



Figuur 5.17: Grafiek van die gemiddelde persentasie fout van die isogrensheuristiek relatief tot die optimale doelfunksiewaarde vir 100 lukraak gegenereerde $K/E/G$ -HTP's teenoor verskillende hoeveelhede funksies in die doelfunksies n . Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is.

Uit die data in Tabel 5-7 en Figuur 5.18 is dit duidelik dat die isogrensheuristiek beduidend korter neem as die ander vertak-en-begrens-algoritmes om die $K/E/G$ -HTP op te los. Dit neem die isogrensheuristiek ongeveer 'n kwart van die tyd om 'n oplossing te kry vir die $K/E/G$ -HTP met 60 funksies in die doelfunksie wat 'n doelfunksiewaarde het wat gemiddeld 0.07% swakker is as die optimale doelfunksiewaarde. Dit sal afhang van hoe dringend 'n oplossing verkry moet word, om te bepaal of 'n 0.07% swakker doelfunksiewaarde in 'n kwart van die tyd meer werd is as die optimale doelfunksiewaarde in die volle tyd.

5.7.2.4 Prestasie van isogrensheuristiek vir $K/E/V$ -HTP's

Stelling 5.1 beteken eintlik dat die isogrensheuristiek 'n algoritme word indien $K/E/G$ -HTP's opgelos word. Dit is daarom nodig dat hierdie heuristiek se prestasie met dié van die breuk- en hellingalgoritme vergelyk word. 'n Eksperiment is gedoen waarin $K/E/G$ -HTP's lukraak gegenereer is met 'n intervallengte van 200 vir die veranderlikes. In hierdie gevalle is die intervallengte waaruit die gemiddelde helling gegenereer word, onafhanklik van die oplossingstyd. Die resultate van hierdie eksperiment word in Tabel 5-8 en Figuur 5.19 gegee.

Dit kan uit Tabel 5-8 en Figuur 5.19 afgelei word dat die oplossingstye van die isogrensheuristiek baie goed vergelyk met die van die breuk- en hellingalgoritme. Nie één van hierdie algoritmes lewer egter beduidend beter resultate as die ander nie. Dit lyk wel asof die oplossingstye vir groter waardes van die aantal funksies in die doelfunksies n daarna neig dat die breukalgoritme die vinnigste, die isogrensheuristiek die tweede vinnigste en die hellingalgoritme die stadigste oplossingstye het.

n	IGH	VSB	IGVSB	IGVB
5	0.1094	0.3281	0.0781	0.0781
10	0.2188	0.4531	0.3438	0.2813
15	0.2344	0.6406	0.4844	0.2656
20	0.4844	0.8906	0.8281	0.5313
25	0.3477	1.9336	1.2617	1.1836
30	0.5938	2.2773	1.4570	1.2031
35	0.6094	2.6094	1.8438	1.5000
40	0.9063	3.4844	2.3594	1.8750
45	1.0000	4.5625	2.7031	2.4375
50	0.9453	5.5508	3.0547	2.6367
55	1.0039	7.5703	4.0352	3.6211
60	1.2969	9.0469	4.4219	4.1875

Tabel 5-7: Die oplossingstye van die isogrensheuristiek en die ander eksakte metodes vir 100 lukraak gegeneerde $K/E/G$ -HTP's teenoor die aantal funksies in die doelfunksie n . Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegeneer. Die veranderlikes se bogrense is lukraak gegeneer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die volgende afkortings word gebruik: IGH = isogrensheuristiek, VSB = vertak-sny-en-begrens-algoritme, IGVSB = isogrens-vertak-sny-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.

5.7.3 Gevolgtrekkings

Alhoewel die isogrens-vertak-en-begrens-algoritme met terugskoue lyk na 'n logiese benadering om die takke in die vertak-en-begrens-boom te verminder, het dit eers na heelwat mislukkings en doodloopstrate duidelik geword dat die isogrens-vertak-en-begrens-algoritme die beste metode is. Die belowendste van hierdie doodloopstrate en mislukkings word hieronder gelys. Let daarop dat hierdie metodes van vertakking nie werk nie.

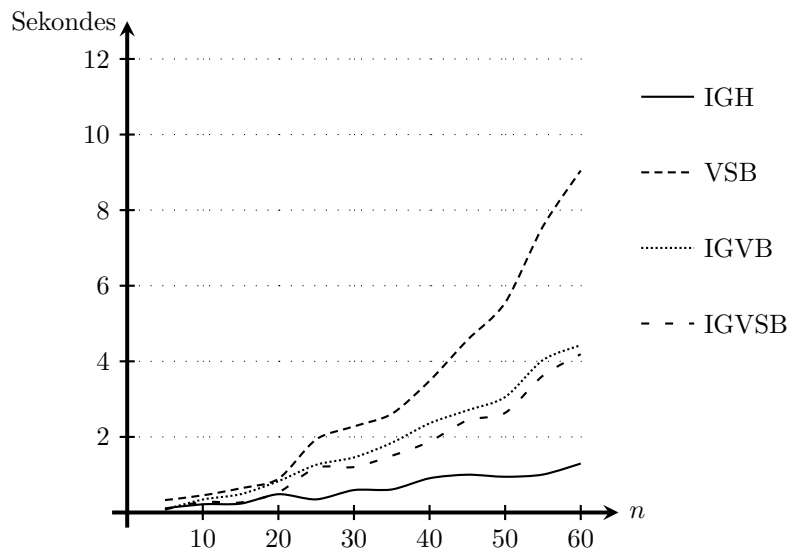
1. 0/1-vertakking op die beste oplossing:

Hierdie metode het daarop neergekom dat die funksie wat in 'n spesifieke subprobleem gekies word om die tob-funksie te wees, vertak word om 'n toedeling óf by sy bogrens óf by sy ondergrens te kry. Hierdie metode kan as 'n goeie heuristiek dien, maar verseker nie altyd die optimale oplossing nie.

2. 0/1-vertakkings:

In 'n ander benadering is elke subprobleem met die iso-ondergrens- en iso-bogrens-prosedure opgelos en die antwoorde met mekaar vergelyk. Indien die antwoorde verskil, word twee funksies se toedelings vertak om die algoritmes in die rigting te dwing waar beide dieselfde oplossing lewer (oftewel dieselfde tob-funksie voorstel). Ook hierdie benadering waarborg nie die optimale oplossing nie.

By die vertak-sny-en-begrens-algoritme is die helling van die tob-funksie eers gebruik om die onder- en bo-snitte te bepaal. Hierdie keuse van helling lei egter tot veel stadiger konvergensie gedurende die vertak-en-begrens-algoritme. Daarom word die resultate vir hierdie metode nie eers hier verskaf nie.



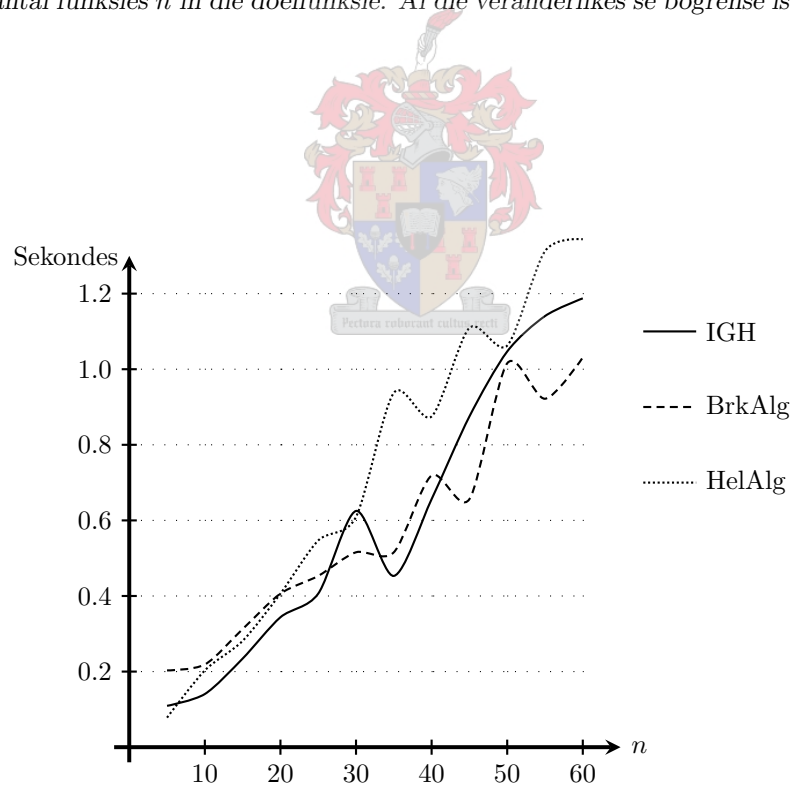
Figuur 5.18: Die grafieke van die oplossingstye van die isogrensheuristiek en die ander eksakte metodes vir 100 lukraak gegeneerde $K/E/G$ -HTP's teenoor die aantal funksies in die doelfunksie n . Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegeneer. Die veranderlikes se bogrense is lukraak gegeneer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die volgende afkortings word gebruik: IGH = isogrensheuristiek, VSB = vertak-sny-en-begrens-algoritme, IGVSB = isogrens-vertak-sny-en-begrens-algoritme en IGVB = isogrens-vertak-en-begrens-algoritme.

In hierdie hoofstuk is algoritmes daargestel wat baie beter oplossings bied as die bestaande metodes wat in die literatuur bekend is. Presies hoeveel beter word in §5.6 en §5.7.2 aangedui.



n	IGH	BrkAlg	HelAlg
5	0.1094	0.2031	0.0781
10	0.1406	0.2188	0.2031
15	0.2344	0.3125	0.2813
20	0.3438	0.4063	0.4063
25	0.4063	0.4531	0.5469
30	0.6250	0.5156	0.6094
35	0.4531	0.5156	0.9375
40	0.6563	0.7188	0.8750
45	0.8750	0.6563	1.1094
50	1.0469	1.0156	1.0625
55	1.1406	0.9219	1.3125
60	1.1875	1.0313	1.3438

Tabel 5-8: Die oplossingstye van die isogrensheuristiek (IGH), die breukalgoritme (BrkAlg) en die hellingalgoritme (HelAlg) vir 100 lukraak gegenereerde $K/E/V$ -HTP's teenoor die aantal funksies n in die doelfunksie. Al die veranderlikes se bogrense is 200 eenhede.



Figuur 5.19: Grafieke van die oplossingstye van die isogrensheuristiek (IGH), die breukalgoritme (BrkAlg) en die hellingalgoritme (HelAlg) vir 100 lukraak gegenereerde $K/E/V$ -HTP's teenoor die aantal funksies n in die doelfunksie. Al die veranderlikes se bogrense is 200 eenhede.



Hoofstuk 6

$K/E, A/G$: Kontinue benadering

Dit is algemeen in wiskunde om eers aannames te maak wat die probleem onder beskouing meer hanteerbaar of eenvoudiger maak. Nadat die eenvoudiger probleem dan goed ondersoek, beskryf en verstaan is, kan van hierdie aannames geleidelik verslap word sodat 'n pad op hierdie wyse oopgewerk word na die meer algemene probleem. In hierdie geval beteken dit dat daar eers na die spesiale geval van een konvekse en een konkawe funksie in die doelfunksie gekyk word, waarna die veralgemening na enige kombinasie van konvekse en konkawe funksies beskou sal word.

6.1 Teoretiese beskouing van een konkawe en een konvekse funksie

Beskou eers die spesiale geval waar daar slegs een konvekse en een konkawe funksie in die doelfunksie is. Gestel $x \in [0, b_x]$ en $y \in [0, b_y]$. Die algemene formulering van die probleem word dan gegee deur

$$\text{maks } z = f(x) + g(y)$$

onderhewig aan

$$\begin{aligned}x + y &= B \\ 0 \leq x &\leq b_x \\ 0 \leq y &\leq b_y.\end{aligned}\tag{6.1}$$

Beskou die probleem soos in (6.1) hierbo gedefinieer. Die tweede orde nodige voorwaardes vanuit Stelling 2.4 vereis dat vir (x^*, y^*) om 'n lokale maksimumpunt te wees, moet daar

'n $\lambda \in \mathbb{R}$ en $\mu_1, \mu_2, \mu_3, \mu_4 \geq 0$ bestaan, sodat

$$x^* + y^* = B \quad (6.2)$$

$$f'(x^*) + \lambda + \mu_1 - \mu_2 = 0 \quad (6.3)$$

$$g'(y^*) + \lambda + \mu_3 - \mu_4 = 0 \quad (6.4)$$

$$\mu_1(x^*) = 0 \quad (6.5)$$

$$\mu_2(b_x - x^*) = 0 \quad (6.6)$$

$$\mu_3(y^*) = 0 \quad (6.7)$$

$$\mu_4(b_y - y^*) = 0 \quad (6.8)$$

en

$$L(x^*, y^*) = \begin{bmatrix} F(x^*) & 0 \\ 0 & G(y^*) \end{bmatrix} \quad (6.9)$$

negatief semidefiniet moet wees op die raakvlak bepaal deur die bindende beperkings, waar $F(x) = \frac{\partial^2 f(x)}{\partial x^2}$ en $G(y) = \frac{\partial^2 g(y)}{\partial y^2}$.

Die tweede orde voldoende voorwaardes vir 'n sterk lokale maksimumpunt is dat (6.2) tot (6.8) moet geld en dat die Lagransiaan in (6.9) negatief definiet moet wees op die raakvlak wat deur die bindende beperkings bepaal word.

Die volgende drie spesiale gevalle kan nou ontstaan:

1. $x + y = B$ is die enigste bindende beperking
2. $x + y = B$ en $0 \leq x \leq b_x$ is bindend ($x = 0$ of $x = b_x$)
3. $x + y = B$ en $0 \leq y \leq b_y$ is bindend ($y = 0$ of $y = b_y$).

Hierdie spesiale gevalle word in die volgende drie onderafdelings beskou.

6.1.1 $x + y = B$ die enigste bindende beperking

Beskou die geval waar $x + y = B$ die enigste bindende beperking is.

Stelling 6.1

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $f'(x^*) = g'(y^*)$, $x^* + y^* = B$ die enigste bindende beperking is en dat $|K_g(y^*)| > |K_f(x^*)|$, dan is (x^*, y^*) 'n sterk relatiewe maksimumpunt vir (6.1).

Bewys: Indien $x + y = B$ die enigste bindende beperking is, impliseer dit dat $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 0$. Hieruit volg die nodige voorwaardes dat die afgeleides van die twee funksies by (x^*, y^*) gelyk aan mekaar moet wees, oftewel

$$f'(x^*) = g'(y^*).$$

Voldoende voorwaardes vir lokale optimaliteit by (x^*, y^*) vereis dat die Legrangiaan $L(x^*, y^*)$ negatief definitief moet wees op die raakvlak M gedefinieer deur

$$\begin{aligned} M &= \left\{ z \mid \frac{\partial(x+y-B)}{\partial x} z_1 + \frac{\partial(x+y-B)}{\partial y} z_2 = 0 \right\} \\ &= \{ z \mid z_1 + z_2 = 0 \} \quad \text{vir alle } z_1 \text{ en } z_2, \end{aligned}$$

by die punt (x^*, y^*) . Vir 'n punt op die raakvlak moet dus volg dat $z_1 = -z_2$. Vir $L(x^*, y^*)$ om negatief definitief op M te wees, moet

$$[z_1 \quad -z_1] \begin{bmatrix} F(x^*) & 0 \\ 0 & G(y^*) \end{bmatrix} \begin{bmatrix} z_1 \\ -z_1 \end{bmatrix} < 0 \quad \text{vir alle } z_1, z_2 \neq 0,$$

waar $F(x) = \frac{\partial^2 f(x)}{\partial x^2}$ en $G(y) = \frac{\partial^2 g(y)}{\partial y^2}$. Uit die definisie van $f(x)$ en $g(y)$ weet ons dat $F(x^*) \geq 0$ en $G(y^*) \leq 0$, sodat volg dat

$$z_1^2 [F(x^*) + G(y^*)] < 0$$

en

$$F(x^*) + G(y^*) < 0$$

mits $F(x^*) < |G(y^*)|$. Hierdie voorwaarde verseker dat $|K_g(y^*)| > |K_f(x^*)|$, wat die stelling bewys. ■

Dit volg uit die vergelykings (6.3) tot (6.8) dat, indien slegs $x + y = B$ in die optimale oplossing bindend is, die hellings van $f(x^*)$ en $g(y^*)$ gelyk aan mekaar moet wees. Hierdie resultaat kan veralgemeen word na meer konkawe en meer konvekse funksies, maar in hierdie geval moet die hellings van al die konkawe funksies wat 'n toedeling tussen hul bo- en ondergrense kry en die helling van die (maksimum een) konvekse funksie wat 'n toedeling tussen sy bo- en ondergrense kry, se helling gelyk wees. Hierdie helling word die **konsensushelling** h_k genoem.

6.1.2 $x + y = B$ en $0 \leq x \leq b_x$ is bindend

Beskou die geval waar die beperkings $x + y = B$ en $x \leq b_x$ bindend is.

6.1.2.1 $x + y = B$ is bindend en $x = b_x$

Stelling 6.2

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $x^* = b_x$ die enigste bindende beperkings is, dan voldoen (x^*, y^*) aan die eerste orde nodige voorwaardes vir 'n relatiewe maksimumpunt vir (6.1) as $f'(x^*) \geq g'(y^*)$.

Bewys: Uit die voorwaarde dat slegs $x + y = B$ en $x = b_x$ bindend is, volg vanuit vergelykings (6.5) tot (6.8) dat $\mu_1 = \mu_3 = \mu_4 = 0$. Omdat $x - b_x = 0$ kan μ_2 enige nie-negatiewe waarde aanneem. Indien hierdie resultaat teruggestel word in vergelykings (6.3)

en (6.4) volg dat

$$\begin{aligned} f'(x^*) + \lambda - \mu_2 &= 0 \\ g'(y^*) + \lambda &= 0, \end{aligned}$$

of

$$f'(x^*) - \mu_2 = g'(y^*).$$

Uit die voorwaarde dat beide $f(x)$ en $g(y)$ monotoon stygend is, weet ons dat $f'(x) \geq 0$ en $g'(y) \geq 0$, wat die stelling bewys. ■

Stelling 6.3

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $x^* = b_x$ die enigste bindende beperkings is en $f'(x^*) \geq g'(y^*)$, dan voldoen (x^*, y^*) aan die tweede orde nodige voorwaardes vir 'n relatiewe maksimumpunt vir (6.1).

Bewys: Volgens Stelling 2.3 is die tweede orde nodige voorwaardes vir optimaliteit dat die punt (x^*, y^*) 'n reguliere punt is en dat die Lagrangiaan negatief semidefiniet moet wees op die raakvlak bepaal deur die aktiewe beperkings. Vir (x^*, y^*) om 'n reguliere punt te wees, moet

$$\begin{aligned} \left[\frac{\partial(x+y-B)}{\partial x}, \frac{\partial(x+y-B)}{\partial y} \right] &= [1, 1] \\ \left[\frac{\partial(b_x-x)}{\partial x}, \frac{\partial(b_x-x)}{\partial y} \right] &= [-1, 0] \end{aligned}$$

by die punt (x^*, y^*) lineêr onafhanklik wees, wat dit wel is. Op die raakvlak is

$$\begin{aligned} \frac{\partial(x+y-B)}{\partial x} z_1 + \frac{\partial(x+y-B)}{\partial y} z_2 &= 0 \\ \frac{\partial(b_x-x)}{\partial x} z_1 + \frac{\partial(b_x-x)}{\partial y} z_2 &= 0 \end{aligned}$$

dit wil sê

$$\begin{aligned} z_1 + z_2 &= 0 \\ -z_1 &= 0 \end{aligned}$$

by die punt (x^*, y^*) , oftewel $z_1 = z_2 = 0$.

Vir die tweede orde nodige voorwaarde moet

$$\underline{z}^T L(x^*, y^*) \underline{z} \leq 0,$$

maar omdat $z_1 = z_2 = 0$ sal

$$\underline{z}^T L(x^*, y^*) \underline{z} = 0,$$

wat beteken dat daar aan die tweede orde nodige voorwaardes voldoen word. ■

Let op dat daar nie in hierdie geval voldoende voorwaardes vir optimaliteit gevind kan word nie.

Afleiding 6.1

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $x^* = b_x$ die enigste bindende beperkings is, dan kan (x^*, y^*) nie aan die tweede orde voldoende voorwaardes vir 'n sterk relatiewe maksimumpunt vir (6.1) voldoen nie.

Beskou nou die geval waar die beperkings $x + y = B$ bindend en $x = 0$ is.

6.1.2.2 $x + y = B$ is bindend en $x = 0$ **Stelling 6.4**

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $x^* = 0$ die enigste bindende beperkings is, dan voldoen (x^*, y^*) aan die eerste orde nodige voorwaardes vir 'n relatiewe maksimumpunt vir (6.1) as $f'(x^*) \leq g'(y^*)$.

Bewys: Uit die voorwaarde dat slegs $x + y = B$ en $x = 0$ bindend is, volg vanuit vergelykings (6.5) tot (6.8) dat $\mu_2 = \mu_3 = \mu_4 = 0$. Omdat $x = 0$ kan μ_1 enige nie-negatiewe waarde aanneem. Indien hierdie resultaat in vergelykings (6.3) en (6.4) gestel word, volg dat

$$\begin{aligned} f'(x^*) + \lambda + \mu_1 &= 0 \\ g'(y^*) + \lambda &= 0, \end{aligned}$$

of

$$f'(x^*) + \mu_1 = g'(y^*).$$

Uit die voorwaarde dat beide $f(x)$ en $g(y)$ monotoon stygend is, weet ons dat $f'(x) \geq 0$ en $g'(y) \geq 0$, wat die stelling bewys. ■

Stelling 6.5

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $x^* = 0$ die enigste bindende beperkings is en $f'(x^*) \leq g'(y^*)$, dan voldoen (x^*, y^*) aan die tweede orde nodige voorwaardes vir 'n relatiewe maksimumpunt vir (6.1).

Bewys: Volgens Stelling 2.3 is die tweede orde nodige voorwaardes vir optimaliteit dat die punt (x^*, y^*) 'n reguliere punt is en dat die Lagrangiaan negatief semidefiniet is op die raakvlak bepaal deur die aktiewe beperkings. Vir (x^*, y^*) om 'n reguliere punt te wees, moet

$$\begin{aligned} \left[\frac{\partial(x + y - B)}{\partial x}, \frac{\partial(x + y - B)}{\partial y} \right] &= [1, 1] \\ \left[\frac{\partial(x - 0)}{\partial x}, \frac{\partial(x - 0)}{\partial y} \right] &= [1, 0] \end{aligned}$$

by die punt (x^*, y^*) lineêr onafhanklik wees, wat dit wel is. Op die raakvlak is

$$\begin{aligned} \frac{\partial(x + y - B)}{\partial x} z_1 + \frac{\partial(x + y - B)}{\partial y} z_2 &= 0 \\ \frac{\partial(x - 0)}{\partial x} z_1 + \frac{\partial(x - 0)}{\partial y} z_2 &= 0 \end{aligned}$$

by die punt (x^*, y^*) , dit wil sê

$$\begin{aligned} z_1 + z_2 &= 0 \\ z_1 &= 0 \end{aligned}$$

oftewel $z_1 = z_2 = 0$.

Vir die tweede orde nodige voorwaarde moet

$$\underline{z}^T L(x^*, y^*) \underline{z} \leq 0,$$

maar omdat $z_1 = z_2 = 0$ sal

$$\underline{z}^T L(x^*, y^*) \underline{z} = 0,$$

wat beteken daar kan slegs aan die tweede orde nodige voorwaardes voldoen word. ■

Let daarop dat daar nie in hierdie geval voldoende voorwaardes vir optimaliteit gevind kan word nie.

Afleiding 6.2

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $x^* = 0$ die enigste bindende beperkings is, dan kan (x^*, y^*) nie aan die tweede orde voldoende voorwaardes vir 'n sterk relatiewe maksimumpunt vir (6.1) voldoen nie.

6.1.3 $x + y = B$ en $0 \leq y \leq b_y$ is bindend

Beskou die geval waar die beperkings $x + y = B$ en $y = b_y$ bindend is.

6.1.3.1 $x + y = B$ is bindend en $y = b_y$

Stelling 6.6

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $y^* = b_y$ die enigste bindende beperkings is, dan voldoen (x^*, y^*) aan die eerste orde nodige voorwaardes vir 'n relatiewe maksimumpunt vir (6.1) as $f'(x^*) \leq g'(y^*)$.

Stelling 6.7

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $y^* = b_y$ die enigste bindende beperkings is en $f'(x^*) \leq g'(y^*)$, dan voldoen (x^*, y^*) aan die tweede orde nodige voorwaardes vir 'n relatiewe maksimumpunt vir (6.1).

Die bewyse vir Stellings 6.6 en 6.7 volg presies soortgelyk aan die bewyse vir Stellings 6.2 en 6.3.

Afleiding 6.3

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $y^* = b_y$ die enigste bindende beperkings is, dan kan (x^*, y^*) nie aan die tweede orde voldoende voorwaardes vir 'n sterk relatiewe maksimumpunt vir (6.1) voldoen nie.

Beskou nou die geval waar die beperkings $x + y = B$ en $y = 0$ bindend is.

6.1.3.2 $x + y = B$ is bindend en $y = 0$ **Stelling 6.8**

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $y^* = 0$ die enigste bindende beperkings is, dan voldoen (x^*, y^*) aan die nodige voorwaardes vir 'n relatiewe maksimumpunt vir (6.1) as $f'(x^*) \geq g'(y^*)$.

Stelling 6.9

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $y^* = 0$ die enigste bindende beperkings is en $f'(x^*) \geq g'(y^*)$, dan voldoen (x^*, y^*) aan die tweede orde nodige voorwaardes vir 'n relatiewe maksimumpunt vir (6.1).

Die bewyse vir Stellings 6.8 en 6.9 volg presies soortgelyk aan die bewyse vir Stellings 6.4 en 6.5.

Afleiding 6.4

Gestel (x^*, y^*) is 'n oplossing van (6.1) sodat $x^* + y^* = B$ en $y^* = 0$ die enigste bindende beperkings is, dan kan (x^*, y^*) nie aan die tweede orde voldoende voorwaardes vir 'n sterk relatiewe maksimumpunt vir (6.1) voldoen nie.

6.1.4 Optimaliteit vir twee funksies

Daar volg nou 'n hele paar gevolgtrekkings uit Stellings 6.2 tot 6.9. Vir enige oplossing (x^*, y^*) moet een van die volgende geld:

1. of $x^* = 0$ en $y^* = 0$
Dit impliseer die triviale geval $B = 0$ en dan is net een oplossing moontlik.
2. of $x^* = b_x$ en $y^* = b_y$
Dit impliseer ook 'n triviale geval, naamlik $B = b_x + b_y$, waar daar ook net een oplossing moontlik is.
3. of $x^* = 0$ en $0 < y^* < b_y$
In hierdie geval kan geen voldoende voorwaardes bepaal word vir globale optimaliteit nie.
4. of $x^* = b_x$ en $0 < y^* < b_y$
In hierdie geval kan geen voldoende voorwaardes bepaal word vir globale optimaliteit nie.
5. of $y^* = 0$ en $0 < x^* < b_x$
In hierdie geval kan geen voldoende voorwaardes bepaal word vir globale optimaliteit nie.
6. of $y^* = b_y$ en $0 < x^* < b_x$
In hierdie geval kan geen voldoende voorwaardes bepaal word vir globale optimaliteit nie.

7. of $0 < x^* < b_x$ en $0 < y^* < b_y$

In hierdie geval is $|K_g(y^*)| > |K_f(x^*)|$ 'n voldoende voorwaarde vir 'n sterk relatiewe maksimumpunt, wat impliseer dat die konkawe funksie meer “bolvormig” as die konvekse funksie by die punt (x^*, y^*) moet wees vir relatiewe optimaliteit.

Die situasies wat hierbo geskep word, het tot gevolg dat 'n algoritme wat die onderstaande doen, al die punte sal dek wat aan die nodige voorwaardes vir optimaliteit voldoen:

1. stel $x = 0$ en los op vir y sodat $x + y = B$;
2. stel $x = b_x$ en los op vir y sodat $x + y = B$;
3. stel $y = 0$ en los op vir x sodat $x + y = B$;
4. stel $x = b_y$ en los op vir x sodat $x + y = B$; en
5. doen 'n lineêre soektog vir die punt(e) waar $f'(x) = g'(y)$ en $x + y = B$ en $f''(x) \geq g''(y)$.

Punte 1 tot 4 hierbo is baie maklik om te bepaal. Punt 5 is ook maklik indien die punt waarna gesoek word, uniek is. Dit is egter problematies indien daar nie aangetoon kan word dat daar net een so 'n punt is nie. In die algemeen is dit definitief moontlik dat daar meer as een so 'n punt kan wees. Dit kan in die algemeen aangetoon word dat so 'n punt nie uniek is nie. In die algemeen kan funksies geskep word wat 'n arbitrêre aantal punte het wat aan die voorwaardes in Punt 5 voldoen, ongeag hoe klein die interval is [28]. 'n Ander uitweg sou dan wees om te bewys dat so 'n punt uniek is vir die families van funksies wat ons in hierdie proefskrif beskou, maar selfs vir hierdie funksies is so 'n punt nie uniek nie. Dit is wel hoogs onwaarskynlik, maar glad nie onmoontlik nie om meer as een so 'n punt te kry waar $f'(x) = g'(y)$ en $x + y = B$ en $f''(x) \geq g''(y)$.

Al wat van die funksies vereis word om meer as een punt te hê wat aan die voorwaardes in Punt 5 voldoen, is dat daar iewers drie punte moet wees, sê (x_1, y_1) , (x_2, y_2) en (x_3, y_3) waarvoor geld dat

$$\begin{aligned}x_1 + y_1 &= B \\f'(x_1) &= g'(y_1) \\f''(x_1) &\leq -g''(y_1)\end{aligned}$$

en

$$\begin{aligned}x_2 + y_2 &= B \\f'(x_2) &= g'(y_2) \\f''(x_2) &\leq -g''(y_2).\end{aligned}$$

Hierdie voorwaarde impliseer natuurlik (volgens die middelwaardstelling en kontinuïteit van f en g) dat daar ook 'n $x_3 \in (x_1, x_2)$ en $y_3 \in (y_1, y_2)$ moet wees sodat

$$\begin{aligned}x_3 + y_3 &= B \\f'(x_3) &= g'(y_3) \\f''(x_3) &\geq -g''(y_3).\end{aligned}$$

Daar is geen rede waarom daar nie drie sulke punte vir die keuse van funksies in hierdie proefskrif kan bestaan nie.

Die probleem van twee funksies kan nog verder vereenvoudig word deur dit te herlei na 'n funksie in net een veranderlike. Alhoewel hierdie formulering eenvoudiger is, bring dit egter geen nuwe insigte na vore nie.

6.1.5 Herlei na 'n funksie in een veranderlike

Die twee funksies $f(x)$ en $g(y)$ in (6.1) kan herlei word na 'n funksie in een veranderlike. Ons weet van die beperking $x + y = B$ dat $y = B - x$. As y dus oral met hierdie waarde vervang word, kan (6.1) herskryf word as

$$\text{maks } h(x)$$

onderhewig aan

$$B - b_y \leq x \leq b_x$$

waar $h(x) = f(x) + g(B - x)$. Die volgende gevalle is moontlik:

1. Die funksie $h(x)$ is streng konveks oor die interval $x \in [B - b_y, b_x]$.
In hierdie geval kan kandidate vir optimale oplossings slegs by eindpunte van die interval bestaan, oftewel by $x = B - b_y$ of $x = b_x$.
2. Die funksie $h(x)$ is streng konkav oor die interval $x \in [B - b_y, b_x]$.
In hierdie geval kan die kandidate vir optimale oplossings by eindpunte van die interval bestaan, dit wil sê by $x = B - b_y$ of $x = b_x$ of by die draaipunt (indien daar een bestaan) van $h(x)$, dit wil sê by die waarde van x waar $h'(x) = 0$.
3. Die funksie $h(x)$ is konveks oor sekere dele en konkav oor ander dele van die interval $x \in [B - b_y, b_x]$.
In hierdie geval kan daar teoreties oneindig baie draaipunte en dus kandidate vir die optimale oplossing in die interval $[B - b_y, b_x]$ bestaan. Die kandidate vir die optimale oplossing kan ook by die eindpunte van die interval, dit wil sê by $x = B - b_y$ of $x = b_x$ bestaan.

6.2 Veralgemening na meer funksies

Indien die funksies in die doelfunksie vermeerder word, kan die probleem geformuleer word as

$$\text{maks } f(\underline{x}) + g(\underline{y}) = \sum_{i=1}^m f_i(x_i) + \sum_{j=1}^n g_j(y_j)$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^m x_i + \sum_{j=1}^n y_j &= B & (6.10) \\ 0 \leq x_i \leq u_i & & i = 1, 2, \dots, m \\ 0 \leq y_j \leq u_j & & j = 1, 2, \dots, n. \end{aligned}$$

Beskou die probleem soos in (6.10) hierbo gedefinieer. Die tweede nodige orde voorwaardes vanuit Stellings 2.3 en 2.4 vereis dat vir $(x_1^*, \dots, x_m^*, y_1^*, \dots, y_n^*)$ om optimaal te wees, moet daar 'n $\lambda \in \mathbb{R}$ en $\mu_1, \dots, \mu_m, \nu_1, \dots, \nu_m, \eta_1, \dots, \eta_n, \omega_1, \dots, \omega_n \geq 0$ bestaan sodat

$$\sum_{i=1}^m x_i^* + \sum_{j=1}^n y_j^* = B \quad (6.11)$$

$$f'(x_i^*) + \lambda + \mu_i - \nu_i = 0 \quad i = 1, \dots, m \quad (6.12)$$

$$g'(y_j^*) + \lambda + \eta_j - \omega_j = 0 \quad j = 1, \dots, n \quad (6.13)$$

$$\mu_i(x_i^*) = 0 \quad i = 1, \dots, m \quad (6.14)$$

$$\eta_j(y_j^*) = 0 \quad j = 1, \dots, n \quad (6.15)$$

$$\nu_i(u_i - x_i^*) = 0 \quad i = 1, \dots, m \quad (6.16)$$

$$\omega_j(u_j - y_j^*) = 0 \quad j = 1, \dots, n \quad (6.17)$$

en

$$L(\underline{x}^*) = \begin{bmatrix} F_1(x_1^*) & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & F_m(x_m^*) & 0 & \dots & 0 \\ 0 & \dots & 0 & G_1(y_1^*) & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & G_n(y_n^*) \end{bmatrix} \quad (6.18)$$

negatief semidefiniet op die raakvlak M vir die bindende beperkings moet wees, waar

$$F_i(x_i) = \frac{\partial^2 f_i(x_i)}{\partial x_i^2} \quad \text{en} \quad G_j(y_j) = \frac{\partial^2 g_j(y_j)}{\partial y_j^2}.$$

Dit beteken dat vir alle waardes van $z_1, z_2, \dots, z_m, w_1, w_2, \dots, w_n \in M$ moet

$$[z_1, \dots, z_m, w_1, \dots, w_n] \begin{bmatrix} F_1(x_1^*) & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & F_m(x_m^*) & 0 & \dots & 0 \\ 0 & \dots & 0 & G_1(y_1^*) & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & G_n(y_n^*) \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_m \\ w_1 \\ \vdots \\ w_n \end{bmatrix} \leq 0. \quad (6.19)$$

Vergelyking (6.19) vereis dus dat

$$z_1^2 F_1(x_1^*) + \dots + z_m^2 F_m(x_m^*) + w_1^2 G_1(y_1^*) + \dots + w_n^2 G_n(y_n^*) \leq 0.$$

Let op dat $M = M_1 \cap M_2 \cap M_3$, sodat volg dat vir $z_i, w_j \in M$ moet

$$z_1 + z_2 + \dots + z_m + w_1 + \dots + w_n = 0 \quad M_1 \quad (6.20)$$

$$\left. \begin{array}{l} z_i = 0 \text{ as } x_i = 0 \\ w_j = 0 \text{ as } y_j = 0 \end{array} \right\} M_2 \quad (6.21)$$

$$\left. \begin{array}{l} z_i = 0 \text{ as } x_i = u_i \\ w_j = 0 \text{ as } y_j = u_j \end{array} \right\} M_3. \quad (6.22)$$

In die veralgemening na meer funksies veroorsaak 'n vermeerdering in die aantal konkawe funksies nie 'n probleem nie. Die rede hiervoor is dat die som van 'n aantal konkawe funksies in hierdie geval beskou kan word as 'n enkele konkawe funksie. Natuurlik sal die enkele konkawe funksie gebou moet word uit al die individuele konkawe funksies deur die optimale toedeling aan die individuele konkawe funksies te bepaal. Die effek van meer konkawe funksies is dus letterlik dat daar 'n klein optimeringstap (waarin die konkawe funksies opgelos word) binne die groter probleem gedoen moet word, maar vanuit die oogpunt vir nodige en voldoende voorwaardes maak dit die probleem niks meer kompleks nie.

Dieselfde geld egter nie by die vermeerdering van die aantal konvekse funksies nie.

Beskou eers die positiewe aspek. Ons het vir $K/E/G$ -HTP's aangetoon dat daar hoogstens een element in versameling \mathcal{K} kan wees. Hierdie resultaat kan veralgemeen word vir $K/E, A/G$ -HTP's. Dit wil sê, indien die doelfunksie uit 'n aantal konkawe en konvekse funksies bestaan, kan daar steeds hoogstens een funksie uit die versameling van konvekse funksie in die versameling \mathcal{K} wees. Anders gestel beteken dit dat daar steeds hoogstens een tob-funksie uit die versameling van konvekse funksies kan kom. Daar kan natuurlik enige aantal funksies uit die versameling konkawe funksies in die versameling \mathcal{K} wees. Stelling 6.10 bewys hierdie resultaat.

Stelling 6.10

'n Optimale oplossing vir 'n $K/E, A/G$ -HTP het hoogstens een konvekse funksie in die versameling \mathcal{K} .

Bewys: Beskou die formulering van 'n $K/E, A/G$ -HTP soos dit in (6.10) gegee is. Hierdie probleem kan geskei word in twee subprobleme (die een bevat al die konkawe (P_1) en die ander al die konvekse funksies(P_2)) en 'n meesterprobleem. Die meesterprobleem word gegee deur

$$\text{maks } z = Z_1 + Z_2$$

onderhewig aan

$$\begin{aligned} B_1 + B_2 &= B \\ B_1, B_2 &\geq 0. \end{aligned} \tag{6.23}$$

Die subprobleem P_1 kan geformuleer word as

$$\text{maks } Z_1 = \sum_{i=1}^m f_i(x_i)$$

onderhewig aan

$$\begin{aligned} \sum_{i=1}^m x_i &= B_1 \\ 0 \leq x_i &\leq u_i \end{aligned} \quad i = 1, 2, \dots, m \tag{6.24}$$

en subprobleem P_2 as

$$\text{maks } Z_2 = \sum_{j=1}^n g_j(y_j)$$

onderhewig aan

$$\begin{aligned} \sum_{j=1}^n y_j &= B_2 \\ 0 \leq y_j &\leq u_j \end{aligned} \quad j = 1, 2, \dots, n. \tag{6.25}$$

Gestel daar bestaan 'n optimale oplossing $(x_1^*, \dots, x_m^*, y_1^*, \dots, y_n^*)$ vir (6.23), dan moet (x_1^*, \dots, x_m^*) 'n optimale oplossing wees vir (6.24) en (y_1^*, \dots, y_n^*) 'n optimale oplossing vir (6.25). Indien dit nie so was nie, sou die betrokke subprobleem met dieselfde hoeveelheid hulpbron 'n beter doelfunksie kon lewer, wat 'n beter oplossing vir die meesterprobleem sou beteken, maar die meesterprobleem is reeds optimaal en 'n beter oplossing is dus nie moontlik nie. Dus moet al twee die subprobleme optimaal wees. As (x_1^*, \dots, x_m^*) 'n optimale oplossing is vir (6.24), geld Stelling 3.1 vir hierdie oplossing. Dit impliseer daar kan hoogstens een konvekse funksie in 'n $K/E, A/G$ -HTP wees wat 'n tob-funksie is. ■

Die waarskynlikheid dat daar meer as een punt is wat 'n relatiewe maksimum vorm, vergroot met 'n toename in die aantal konvekse funksies. Om te verseker dat daar meer

as een relatiewe maksimumpunt is, moet daar minstens twee punte wees waarvoor geld dat (onthou dat alle $g''(\cdot) < 0$)

$$\begin{aligned} \sum_{i=1}^m x_i + \sum_{j=1}^n y_j &= B, \\ f'_i(x_i) &= g'_j(y_j) \quad \forall i, j \in \mathcal{K} \text{ en} \\ \sum_{\forall i \in \mathcal{K}} z_i^2 f''_i(x_i) &\leq \sum_{\forall j \in \mathcal{K}} w_j^2 g''_j(y_j) \quad \forall z_i, w_j \in M. \end{aligned}$$

Indien daar meer as een so 'n punt is, moet daar tussen hierdie punte 'n punt wees wat 'n relatiewe minimum aanneem. Dit beteken tussen die punte wat aan die eienskappe hierbo voldoen, moet daar 'n punt wees sodat

$$\begin{aligned} \sum_{i=1}^m x_i + \sum_{j=1}^n y_j &= B, \\ f'_i(x_i) &= g'_j(x_j) \quad \forall i, j \in \mathcal{K} \text{ en} \\ \sum_{\forall i \in \mathcal{K}} z_i^2 f''_i(x_i) &\geq \sum_{\forall j \in \mathcal{K}} w_j^2 g''_j(x_j) \quad \forall z_i, w_j \in M. \end{aligned}$$

Dit impliseer dat daar minstens twee punte moet wees waarin

$$z_i^2 f''_i(x_i) = \sum_{\forall j \in \mathcal{K}} w_j^2 g''_j(y_j), \quad (6.26)$$

met hoogstens een $z_i \neq 0$. 'n Toename in die aantal konvekse funksies en gevolglik 'n toename in z_i 's veroorsaak 'n groter keuse van funksies aan die linkerkant van (6.26) om hierdie vergelyking waar te maak.

Met die eerste oogopslag kan dit lyk of 'n logiese benadering tot hierdie probleem sou wees om net om die beurt (met ander woorde een op 'n slag) die konvekse funksies 'n kans te gee om die enkele tob-funksie te wees. Die res van die konvekse funksies kan dan as hul lineêre omsluiting (wat impliseer hulle is lyne en dus konkav) saam met die konkawe deel van die probleem gereken word. Dit is egter nie so eenvoudig nie, want dit gebeur dikwels dat een van hierdie "lyne" (wat slegs 'n toedeling op hul bo- of ondergrens mag kry) 'n toedeling tussen hul bo- en ondergrense kry. Dit impliseer weer dat 'n 0/1 vertakkingsalgoritme gebruik moet word. Hierdie vertakking, saam met die feit dat elke konvekse funksie om die beurt as 'n tob-funksie beskou moet word, maak die oplossingstyd van hierdie probleem onaanvaarbaar lank. Die oplossingstyd vir een konvekse funksie saam met die 0/1 vertakking neem min of meer dieselfde tyd, maar meestal langer, as 'n volledige oplossing met gewone vertak-en-begrens-algoritme wat in die volgende hoofstuk voorgestel word, en hierdie oplossing moet soveel keer as wat daar konvekse funksies is, opgelos word. So 'n algoritme is dus reeds op hierdie punt gedoem om swakker resultate as dié in die volgende hoofstuk te lewer.

Die kode vir die algoritme om die 0/1-vertakking te hanteer, word ook op die CD wat aan die einde van hierdie proefskrif ingebind is, gegee. 'n Nuwe algoritme is geskep om

die baie effektiewe Horowitz-Shani-algoritme [24] vir die 0/1-knapsakprobleem aan te pas om die konkawe deel van die probleem te hanteer. Omdat hierdie algoritme so swak vaar, word dit nie hier weergegee nie, maar 'n geïnteresseerde leser kan dit op die CD gaan naslaan.

Die hoofrede vir die swak oplossingstye van die 0/1-vertakking is die byvoeging van die konkawe funksies. Dit het tot gevolg dat baie meer 0/1-kombinasies moontlik is omdat van die hulpbron van/na die konkawe deel van die probleem oorgedra kan word om as't ware meer kombinasies van 0/1-vertakkings te "finansier" oftewel moontlik te maak.

6.3 Gevolgtrekkings

Die kontinue benadering het aan die begin na 'n baie goeie opsie gelyk, maar nadat die wiskunde daarvan deurtrap is, is hierdie benadering om twee redes nie 'n geslaagde een nie:

1. Die uniekheid van oplossings:

Dit kan wiskundig aangetoon word dat daar 'n groot (om die waarheid te sê enige arbitrêr gekose [28]) aantal punte kan wees wat voldoen aan die voorwaardes dat $\sum x_i = B$, $f'_i = g'_j(x_j) \forall i, j \in \mathcal{K}$, en $\sum f''_i(x_i) \geq \sum g''_j(y_j) \forall i, j \in \mathcal{K}$. Hierdie resultaat maak dit prakties onmoontlik om na al die punte wat aan die eerste- en tweede orde nodige voorwaardes voldoen, te gaan soek. Dit skep wel die moontlikheid om na die eerste punt te soek wat aan hierdie voorwaarde voldoen en net daar te stop en die tegniek as 'n heuristiek aan te bied, maar selfs dan gaan die oplossingstyd daarvan nie vergelykbaar wees met die eksakte oplossing wat die vertak-en-begrens-algoritmes wat in die volgende hoofstuk aangebied word, lewer nie, omdat die res van die probleem steeds met 0/1-vertakkings opgelos moet word.

2. Die 0/1-vertakkings wat nodig is, is tydrowend om op te los en kan nie kompeteer met die vertak-en-begrens tipe algoritmes wat in die volgende hoofstuk volg nie.

Die gevolgtrekking is dus dat 'n kontinue benadering tot die oplossing van die $K/E, A/G$ -HTP nooit sal kan kompeteer met die vertak-en-begrens-algoritmes se oplossingstye nie. Die hoofrede hiervoor is dat daar nie voldoende voorwaardes vir oplossings van hierdie tipe probleem bestaan nie. Alhoewel daar in sommige gevalle voldoende voorwaardes vir 'n plaaslike optimum is, moet daar steeds deur al die moontlike punte wat aan die nodige voorwaardes voldoen, gesoek word na kandidaatoplossings en hierdie soektogte is te tydrowend relatief tot vertak-en-begrens-algoritmes.

Hoofstuk 7

$K/E, A/G$:

Vertak-en-begrens-algoritmes

In hierdie hoofstuk word die vertak-en-begrens-algoritmes wat in Hoofstuk 5 ingevoer is, aangepas sodat dit die $K/E, A/G$ -HTP kan hanteer.

7.1 KL-algoritme (vertak-en-begrens-algoritme)

In al die algoritmes wat in hierdie hoofstuk voorgestel word, moet die lineêre omsluiting van die konvekse funksies tesame met 'n aantal konkawe funksies opgelos word. Tegnies gesproke is die lineêre omsluitings van die konvekse funksies ook konkawe funksies, wat eintlik impliseer dat hierdie probleem met die standaard algoritmes vir nielineêre programmering opgelos kan word. Aanvanklik is hierdie benadering dan ook gevolg, maar indien die isogrens-vertak-en-begrens-algoritme vir die $K/E, A/G$ -HTP uitgebrei word, skiet hierdie algoritmes tekort met betrekking tot oplossingstyd. Die rede hiervoor is dat die bestaande algoritmes nie maklik aangepas kan word om die nuwe optimum te vind indien net een lyn (omsluiting van 'n konvekse funksie) verander nie. Daarom het die behoefte ontstaan om 'n algoritme te skep wat weer eens die struktuur van die probleem uitbuit en terselfdertyd maklik nuwe optimums kan vind, mits net een lyn verander. Die algoritme wat vir hierdie doel ontwikkel is, is die KL-algoritme. Dit word so genoem omdat dit met behulp van vertak-en-begrensing die optimale oplossing kan bepaal vir 'n HTP wat net **K**onkawe funksies en **L**yne (lineêre omsluiting van konvekse funksies) in die doelfunksie het.

Aanvaar dat al die lyne van hoogste tot laagste gemiddelde helling gerangskik is. Die werking van die algoritme berus dan daarop dat hierdie hellings een vir een as die konsensus-helling beskou word. Omdat die konkawe funksies en hul afgeleides streng stygende funksies is, kan 'n inverse funksie vir al hierdie afgeleide funksies bepaal word. Hierdie inverse funksies se toevoer is 'n helling en sy afvoer is die x -waarde waarby die funksie hierdie helling het. Op dié manier kan die totale hoeveelheid hulpbron van 'n oplossing wat met hierdie konsensus-helling ooreenstem, feitlik oombliklik (met 'n enkele vir-lus

oor al die konkawe funksies) bereken word. Die funksies waarvoor geld dat $f'_i(u_i) \geq h_k$ word op hul bogrense vasgestel en die funksie waarvoor geld dat $f'_i(l_i) \leq h_k$ word by hul ondergrense vasgestel sodat die probleem aan die voldoende voorwaardes vir optimaliteit kan voldoen. Indien die totale hoeveelheid hulpbron kleiner is as die hoeveelheid hulpbron gebruik vir die optimale oplossing by h_k dan moet h_k vergroot en die proses herhaal word. 'n Newton-tipe soekalgoritme kan dan gebruik word om die regte konsensushelling te bereken, mits dit tussen die hellings van twee opeenvolgende lyne val. Die notasie wat in die KL-algoritme gebruik word, word aan die begin van die proefskrif verskaf en pseudokode vir die algoritme volg in Algoritme 7.1¹.

Algoritme 7.1 (KL (vertak-en-begrens))

Prosedure KL(HTP)

$k \leftarrow 0$

$A_p \leftarrow 1$

$A_u \leftarrow 0.0001$

herhaal

$k = k + 1$

$B_{kv} = 0$

vir $i \leftarrow 1$ **tot** A_{kv} **doen**

$B_{kv} \leftarrow B_{kv} + l_{[i]}^k$

$x_{[i]}^t \leftarrow l_{[i]}^k$

end vir

vir $i \leftarrow 1$ **tot** A_{kv} **doen**

$H \leftarrow g h_{[i]}^k$

$B_t \leftarrow 0$

vir $j \leftarrow 1$ **tot** A_{kk} **doen**

$B_t \leftarrow B_t + f_{[i]}^{-1}(H)$

end vir

as $B_t + B_{kv} > B$ **dan**

$B_{kk} \leftarrow B - B_{kv}$

$T \leftarrow 0$

$M_i \leftarrow KLN$

$M_a \leftarrow GRT$

herhaal

$T = T + 1$

$B_t = 0$

$H = (M_i + M_a)/2$

vir $\ell \leftarrow A_{kv} + 1$ **tot** A_{kk} **doen**

$x_{[\ell]}^k \leftarrow f_{[\ell]}^{-1}(H)$

$B_t \leftarrow B_t + x_{[\ell]}^k$

end vir

as $B_t > B_{kk}$ **dan**

▷ Los subprobleem k op



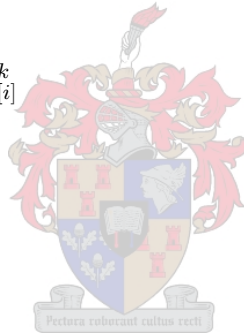
▷ Bereken oplossing en stop

¹Indien 'n indeks in vierkantige hakies gegee word, beteken dit dat die funksies, veranderlikes ens. waarvoor hierdie indeks strek, gesorteer is van hoogste tot laagste g_h .

```

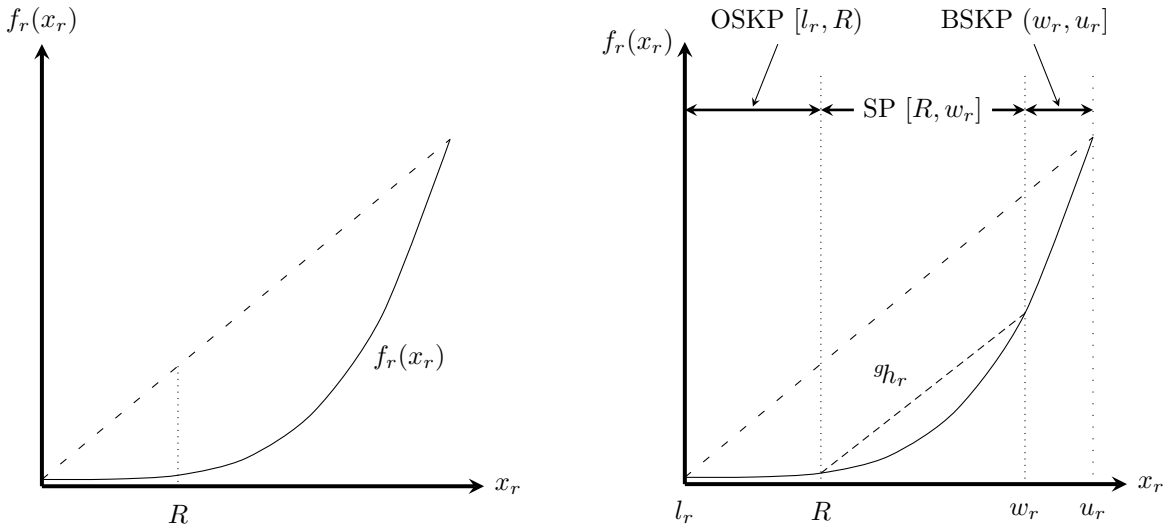
     $M_i \leftarrow H$ 
anders
     $M_a \leftarrow H$ 
end as
as  $T > 1000$  dan
    ontsnap vor
end as
totdat  $|B_{kk} - B_t| < A_u$ 
ontsnap vir
anders as  $B_t + B_{kv} + u_{[i]}^k - l_{[i]}^k > B$  dan
     $\triangleright$  Bereken oplossing en vertak
as  $D_L(\underline{x}_t) > D_b$  dan
     $\triangleright$  Vertak
     $A_p \leftarrow A_p + 2$ 
     $H^{A_p} \leftarrow H^k$ 
     $H^{A_p-1} \leftarrow H^k$ 
     $u_{[i]}^{A_p} \leftarrow B - B_t - B_{kv} - l_{[i]}^k$ 
     $l_{[i]}^{A_p-1} \leftarrow B - B_t - B_{kv} - l_{[i]}^k$ 
end as
ontsnap vir
anders
     $B_{kv} \leftarrow B_{kv} + u_{[i]}^k - l_{[i]}^k$ 
     $x_{[i]}^t \leftarrow u_{[i]}^k$ 
end as
end vir
as  $D_w \underline{x}_t > D_b$  dan
     $\triangleright$  Stoor die beste oplossing
     $D_b \leftarrow D_w \underline{x}_t$ 
     $\underline{x}_b \leftarrow \underline{x}_t$ 
end as
totdat  $K \geq Apr$ 
end Procedure

```



7.2 SKL-algoritme (vertak-sny-en-begrens)

Indien die konkawe funksies saam met die konvekse funksies in die doelfunksie voorkom, geld die afsnyding van dele van die toelaatbare gebied nie meer soos wat ons in Afdeling 5.4 afgelei het nie. Die rede hiervoor is dat Stellings 5.2 en 5.3 nie meer geld met die insluiting van die konkawe funksie nie. Hierdie stellings kan nie sonder meer hier toegepas word nie, omdat daar in die hierdie geval in werklikheid twee dele van die hulpbron bestaan. Die een deel is die deel wat aan die konkawe funksies toegedeel word, terwyl die tweede deel die res van die hulpbron is wat aan die konvekse funksie toegedeel word. Die oomblik wanneer die helling van die konvekse funksie verander soos wat die geval is by die afsnyding in Stellings 5.2 en 5.3, verander die verhouding van hierdie twee dele van die hulpbron. Dit het tot gevolg dat die deel wat aan die tob-funksie toegedeel word, verander en die



(a) Die tob-funksie (f_r) van die ouerprobleem wat in die optimale oplossing van die lineêre omsluiting 'n toedeling van $x_r = R$ kry.

(b) Skep twee subprobleme, naamlik die OSKP en die BSKP wat verder ondersoek moet word. Die derde subprobleem SP se optimale oplossing is reeds bekend en hoef dus nie verder ondersoek te word nie.

Figuur 7.1: Skematiese voorstelling van die geval waar $R < f_r^{-1}(g_{h_r})$.

huidige optimale oplossing nie optimaal bly vir die subprobleem nie. Dit lei weer daartoe dat dié deel van die toelaatbare gebied wat afgesny kan word, nie so groot is as wat die geval in Hoofstuk 5 was nie. In hierdie geval kan die snit op twee maniere bereken word, afhangend van die waarde van R . Die snit wanneer $R < f_r^{-1}(g_{h_r})$, word in Figuur 7.1 gegee en die snit wanneer $R > f_r^{-1}(g_{h_r})$ word in Figuur 7.2 gegee. Geen snit is moontlik indien $R = f_r^{-1}(g_{h_r})$ nie. Die ouerprobleem se konvekse tob-funksie word onderskeidelik in Figuur 7.1(a) en Figuur 7.2(a) gegee en die OSKP, SP en BSKP word in onderskeidelik Figuur 7.1(b) en Figuur 7.2(b) gegee.

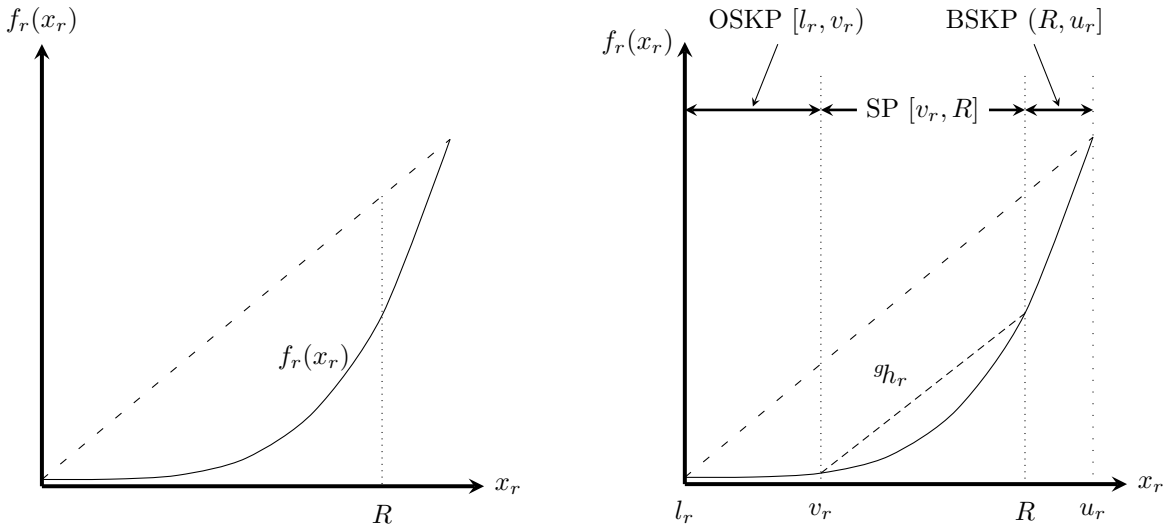
Dit is dus te wagte, en inderdaad ook die geval, dat die besparing aan oplossingstyd nie so dramaties is nie. In baie gevalle oortref die ekstra tyd om die snit te bereken die besparing wat bewerkstellig word deurdat minder subprobleme opgelos moet word. In sulke gevalle is die berekeningstyd selfs langer as wat die geval is sonder die snyding. 'n Algoritme wat van die snyding van die toelaatbare gebied gebruik maak, is die SKL-algoritme wat in Algoritme 7.2 gegee word. Die SKL-algoritme word so genoem omdat dit met behulp van vertak-Sny-en-begrensing die optimale oplossing kan bepaal vir 'n HTP wat net Konkawe funksies en Lyne (lineêre omsluiting van konvekse funksies) in die doelfunksie het. Die werking van hierdie algoritme is presies soos dié van die KL-algoritme, behalwe dat dele van die toelaatbare gebied gesny word in die subprobleme wat geskep word.

Algoritme 7.2 (SKL (vertak-en-begrens))

Prosedure SKL(HTP)

$k \leftarrow 0$

$A_p \leftarrow 1$



(a) Die tob-funksie (f_r) van die ouerprobleem wat in die optimale oplossing van die lineêre omsluiting 'n toedeling van $x_r = R$ kry.

(b) Skep twee subprobleme, naamlik die OSKP en die BSKP wat verder ondersoek moet word. Die derde subprobleem SP se optimale oplossing is reeds bekend en hoef dus nie verder ondersoek te word nie.

Figuur 7.2: Skematiese voorstelling van die geval waar $R > f_r^{-1}(g_h_r)$

$A_u \leftarrow 0.0001$

herhaal

$k = k + 1$

$B_{kv} = 0$

vir $i \leftarrow 1$ **tot** A_{kv} **doen**

$B_{kv} \leftarrow B_{kv} + l_{[i]}^k$

$x_{[i]}^t \leftarrow l_{[i]}^k$

end vir

vir $i \leftarrow 1$ **tot** A_{kv} **doen**

$H \leftarrow g_{h_{[i]}}^k$

$B_t \leftarrow 0$

vir $j \leftarrow 1$ **tot** A_{kk} **doen**

$B_t \leftarrow B_t + f_{[i]}^{-1}(H)$

end vir

as $B_t + B_{kv} > B$ **dan**

$B_{kk} \leftarrow B - B_{kv}$

$T \leftarrow 0$

$M_i \leftarrow KLN$

$M_a \leftarrow GRT$

herhaal

$T = T + 1$

$B_t = 0$

$H = (M_i + M_a)/2$

vir $\ell \leftarrow A_{kv} + 1$ **tot** A_{kk} **doen**



▷ Los subprobleem k op

▷ Bereken oplossing en stop

```

     $x_{[\ell]}^k \leftarrow f_{[\ell]}^{-1}(H)$ 
     $B_t \leftarrow B_t + x_{[\ell]}^k$ 
end vir
as  $B_t > B_{kk}$  dan
     $Mi \leftarrow H$ 
anders
     $Ma \leftarrow H$ 
end as
as  $T > 1000$  dan
    ontsnap vir
end as
totdat  $|B_{kk} - B_t| < A_u$ 
ontsnap vir
anders as  $B_t + B_{kv} + u_{[i]}^k - l_{[i]}^k > B$  dan  $\triangleright$  Bereken oplossing en vertak
as  $D_L(x_t) > D_b$  dan  $\triangleright$  Vertak
     $A_x \leftarrow B - B_t - B_{kv} - l_{[i]}^k$ 
     $A_p \leftarrow A_p + 2$ 
     $H^{A_p} \leftarrow H^k$ 
     $H^{A_p-1} \leftarrow H^k$ 
as  $f'_{[i]}(A_x) > g_{h_{[i]}}$  dan  $\triangleright$  Sny na onder
     $l_{[i]}^{A_p} \leftarrow B - B_t - B_{kv} - l_{[i]}^k$ 
     $T \leftarrow 0$   $\triangleright$  Bereken nuwe bogrens
     $Mi \leftarrow l_{[i]}^k$ 
     $Ma \leftarrow A_x$ 
herhaal
     $T = T + 1$ 
     $H = (Ma + Mi)/2$ 
     $F_w = f_{[i]}(H)$ 
     $L_w = f_{[i]}(A_x) - g_{h_{[i]}}^k \times (A_x - H)$ 
as  $F_w > L_w$  dan
     $Mi = H$ 
anders
     $Ma = H$ 
end as
totdat  $|F_w - L_w| < 0.0001$ 
     $u_{[i]}^{A_p-1} \leftarrow H$ 
anders as  $f'_{[i]}(A_x) > g_{h_{[i]}}$  dan  $\triangleright$  Sny na bo
     $u_{[i]}^{A_p-1} \leftarrow A_x$ 
     $T \leftarrow 0$   $\triangleright$  Bereken nuwe ondergrens
     $Mi \leftarrow A_x$ 
     $Ma \leftarrow u_{[i]}^k$ 
herhaal
     $T \leftarrow T + 1$ 
     $H = (Ma + Mi)/2$ 

```



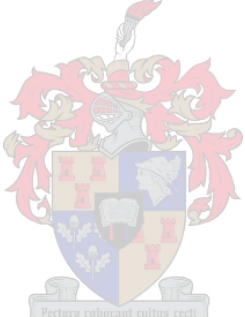
```


$$F_w = f_{[i]}(H)$$


$$L_w = f_{[i]}(A_x) + g_{[i]}^k \times (H - A_x)$$

as  $F_w > L_w$  dan
     $Ma \leftarrow H$ 
anders
     $Mi \leftarrow H$ 
end as
totdat  $|F_w - L_w| < 0.0001$ 
 $l_{[i]}^{Ap} \leftarrow H$ 
anders
 $l_{[i]}^{Ap} \leftarrow B - B_t - B_{kv} - l_{[i]}^k$ 
 $u_{[i]}^{Ap-1} \leftarrow B - B_t - B_{kv} - l_{[i]}^k$ 
end as
end as
ontsnap vir
anders
 $B_{kv} \leftarrow B_{kv} + u_{[i]}^k - l_{[i]}^k$ 
 $x_{[i]}^t \leftarrow u_{[i]}^k$ 
end as
end vir
as  $D_w \underline{x}_t > D_b$  dan
 $D_b \leftarrow D_w \underline{x}_t$ 
 $\underline{x}_b \leftarrow \underline{x}_t$ 
end as
totdat  $K \geq Apr$ 
end Prosedure

```



▷ Stoor die beste oplossing

7.3 IKL-algoritme (isogrens-vertak-en-begrens)

Die filosofie agter die IKL-algoritme² is soortgelyk aan die isogrens-vertak-en-begrens-algoritme wat in Hoofstuk 5 voorgestel word. Dit wil sê hierdie algoritme bestaan ook uit twee dele. Die een prosedure los 'n subprobleem se ondergrensnageslag (OKL-prosedure³) en die ander 'n subprobleem se bogrensnageslag (BKL-prosedure⁴) in een iterasie op. Die notasie wat in die pseudokode van die IKL-algoritme gebruik word, word aan die begin van die proefskrif verskaf. Die pseudokode vir die IKL-algoritme word in Algoritme 7.3 gegee.

²Isogrens-vertak-en-begrens-algoritme vir 'n HTP wat net **K**onkawe funksies en **L**yne (lineêre omsluiting van konvekse funksies) in die doelfunksie het.

³Los die **O**ndergrensnageslag van 'n subprobleem wat net **K**onkawe funksies en **L**yne (lineêre omsluiting van konvekse funksies) in die doelfunksie het, op.

⁴Los die **B**ogrensnageslag van 'n subprobleem wat net **K**onkawe funksies en **L**yne (lineêre omsluiting van konvekse funksies) in die doelfunksie het, op.

Algoritme 7.3 (IKL (vertak-en-begrens))

Prosedure IKL(HTP)

$P \leftarrow 0$

$D \leftarrow 0$

$P_{ap} \leftarrow 1$

$D_{ap} \leftarrow 0$

herhaal

▷ Los subprobleme op

as $P \leq P_{ap}$ **dan**

Roep: IOKL(H^P) ▷ Los die P -de subprobleem se ondergrensnageslag op

$P \leftarrow P + 1$

end as

as $D > 0$ **en** $D < D_{ap}$ **dan**

$D \leftarrow D + 1$

Roep: IBKL(H^D) ▷ Los die D -de subprobleem se bogrensnageslag op

end as

as $P \geq P_{ap}$ **dan**

$P_s \leftarrow 1$

anders

$P_s \leftarrow 0$

end as

as $D \geq D_{ap}$ **dan**

$D_s \leftarrow 1$

anders

$D_s \leftarrow 0$

end as

totdat $P_s = 1$ **en** $D_s = 1$

end Prosedure

Prosedure OKL(H^P)

vir $i \leftarrow 1$ **tot** A_{kv} **doen**

$B_{kv} \leftarrow B_{kv} + l_{[i]}^k$

$x_{[i]}^t \leftarrow l_{[i]}^k$

end vir

vir $i \leftarrow 1$ **tot** A_{kv} **doen**

herhaal

$H \leftarrow g_{h_{[i]}^k}$

$B_t \leftarrow 0$

vir $j \leftarrow A_{kv} + 1$ **tot** $A_{kv} + A_{kk}$ **doen**

$B_t \leftarrow B_t + f_{[i]}^{-1}(H)$

end vir

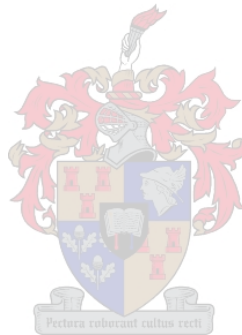
as $B_t + B_{kv} > B$ **dan**

▷ Bereken oplossing en stop

$B_{kk} \leftarrow B - B_{kv}$

$T \leftarrow 0$

$M_i \leftarrow KLN$




```

     $Ma \leftarrow GRT$ 
herhaal
     $T = T + 1$ 
     $B_t = 0$ 
     $H = (Mi + Ma)/2$ 
vir  $\ell \leftarrow A_{kv} + 1$  tot  $A_{kk}$  doen
     $x_{[\ell]}^k \leftarrow f_{[\ell]}^{-1}(H)$ 
     $B_t \leftarrow B_t + x_{[\ell]}^k$ 
end vir
as  $B_t > B_{kk}$  dan
     $Mi \leftarrow H$ 
anders
     $Ma \leftarrow H$ 
end as
as  $T > 1000$  dan
    ontsnap vir
end as
totdat  $|B_{kk} - B_t| < A_u$ 
ontsnap vir
anders as  $B_t + B_{kv} + u_{[i]}^k - l_{[i]}^k > B$  dan  $\triangleright$  Bereken oplossing en vertak
as  $D_L(x_t) > D_b$  dan  $\triangleright$  Skep subprobleem en plaas dit in  $\mathcal{D}$ 
     $D_{ap} \leftarrow D_{ap} + 1$ 
     $H^{D_{ap}} \leftarrow H^P$ 
     $l_{[i]}^{D_{ap}} \leftarrow B - B_t - B_{kv} - l_{[i]}^P$ 
     $u_{[i]}^P \leftarrow B - B_t - B_{kv} - l_{[i]}^P$   $\triangleright$  Pas  $H^P$  aan
anders
    ontsnap procedure
end as
anders
     $B_{kv} \leftarrow 0$ 
vir  $i \leftarrow 1$  tot  $A_{kv}$  doen  $\triangleright$  Los subprobleem  $k$  op
     $B_{kv} \leftarrow B_{kv} + l_{[i]}^k$ 
     $x_{[i]}^t \leftarrow l_{[i]}^k$ 
end vir
end as
totdat
as  $D_w \underline{x}_t > D_b$  dan  $\triangleright$  Stoor die beste oplossing
     $D_b \leftarrow D_w \underline{x}_t$ 
     $\underline{x}_b \leftarrow \underline{x}_t$ 
end as
end vir
end Procedure
Procedure BKL( $H^D$ )
    vir  $i \leftarrow 1$  tot  $A_{kv}$  doen


```

```

 $B_{kv} \leftarrow B_{kv} + l_{[i]}^k$ 
 $x_{[i]}^t \leftarrow l_{[i]}^k$ 
end vir
vir  $i \leftarrow A_{kv}$  tot 1 doen stap -1
  herhaal
     $H \leftarrow g_{h_{[i]}^k}$ 
     $B_t \leftarrow 0$ 
    vir  $j \leftarrow A_{kv} + 1$  tot  $A_{kv} + A_{kk}$  doen
       $B_t \leftarrow B_t + f_{[i]}^{-1}(H)$ 
    end vir
    as  $B_t + B_{kv} < B$  dan
       $B_{kk} \leftarrow B - B_{kv}$ 
       $T \leftarrow 0$ 
       $Mi \leftarrow KLN$ 
       $Ma \leftarrow GRT$ 
      herhaal
         $T = T + 1$ 
         $B_t = 0$ 
         $H = (Mi + Ma)/2$ 
        vir  $\ell \leftarrow A_{kv} + 1$  tot  $A_{kk}$  doen
           $x_{[\ell]}^k \leftarrow f_{[\ell]}^{-1}(H)$ 
           $B_t \leftarrow B_t + x_{[\ell]}^k$ 
        end vir
        as  $B_t > B_{kk}$  dan
           $Mi \leftarrow H$ 
        anders
           $Ma \leftarrow H$ 
        end as
        as  $T > 1000$  dan
          ontsnap vir
        end as
        totdat  $|B_{kk} - B_t| < A_u$ 
        ontsnap vir
        anders as  $B_t + B_{kv} + u_{[i]}^k - l_{[i]}^k < B$  dan
           $D_L(x_t) > D_b$  dan
             $P_{ap} \leftarrow P_{ap} + 1$ 
             $H^{P_{ap}} \leftarrow H^D$ 
             $u_{[i]}^{P_{ap}} \leftarrow B - B_t - B_{kv} - l_{[i]}^D$ 
             $u_{[i]}^D \leftarrow B - B_t - B_{kv} - l_{[i]}^D$ 
          anders
            ontsnap prosedure
          end as
        anders
           $B_{kv} \leftarrow 0$ 

```

▷ Bereken oplossing en stop



▷ Bereken oplossing en vertak

▷ Skep subprobleem en plaas dit in \mathcal{P}

▷ Pas H^D aan

```

vir  $i \leftarrow 1$  tot  $A_{kv}$  doen
     $B_{kv} \leftarrow B_{kv} + l_{[i]}^k$ 
     $x_{[i]}^t \leftarrow l_{[i]}^k$ 
end vir
end as
totdat
as  $D_w \underline{x}_t > D_b$  dan
     $D_b \leftarrow D_w \underline{x}_t$ 
     $\underline{x}_b \leftarrow \underline{x}_t$ 
end as
end vir
end Procedure

```

▷ Los subprobleem k op

▷ Stoor die beste oplossing

A_{kv}	Algoritme		
	KL	SKL	IKL
5	0.0625	0.2031	0.0469
10	0.0781	0.1875	0.1250
15	0.1250	0.2813	0.2656
20	0.1563	0.2969	0.1719
25	0.3438	0.4063	0.2656
30	0.3281	0.5000	0.2344
35	0.5313	0.5469	0.2344
40	0.7656	0.8906	0.5625
45	0.9219	0.8906	0.7031
50	1.3906	1.4375	0.9219
55	2.2813	2.2125	1.7031

Tabel 7-1: Die totale oplossingstyd in sekondes van 100 lukraak gegenereerde K/E , A/G -HTP's wat elkeen 60 funksies in die doelfunksie het, waarvan A_{kv} konvekse funksies is. Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die volgende afkortings word gebruik: KL = vertak-en-begrens-algoritme vir K/E , A/G -HTP's, SKL = vertak-sny-en-begrens-algoritme vir K/E , A/G -HTP's, IKL = isogrens-vertak-en-begrens-algoritme vir K/E , A/G -HTP's.

7.4 Samevatting

Soos voorafgaande hoofstukke bestaan die samevatting uit drie onderafdelings, naamlik 'n onderafdeling oor die metodologie, 'n onderafdeling met resultate en 'n onderafdeling waarin gevolgtrekkings verskaf word oor die resultate.

7.4.1 Metodologie

Dieselfde metodologie as in §3.6.1 en §4.3.1 is gebruik. Die konkawe en konvekse funksies word op soortgelyke wyse gegenereer. Die gebruiker kan spesifiseer hoeveel konkawe en hoeveel konvekse funksies onderskeidelik gegenereer word. In elke datastel is daar A_{kv} konvekse funksies en $60 - A_{kv}$ konkawe funksies gegenereer. Die enkele gevalle waar waar daar wel afgewyk word van hierdie metodologie, word die afwyking in die betrokke byskrif van die tabel of grafiek verskaf.

Die probleme word met dieselfde rekenaar as in §3.6.1 opgelos. Die rekenaarkode vir die algoritmes wat in hierdie hoofstuk aangebied word, kan op die CD agterin hierdie proefskrif in die lêer met lêernaam `keag.xls` gevind word.

7.4.2 Resultate

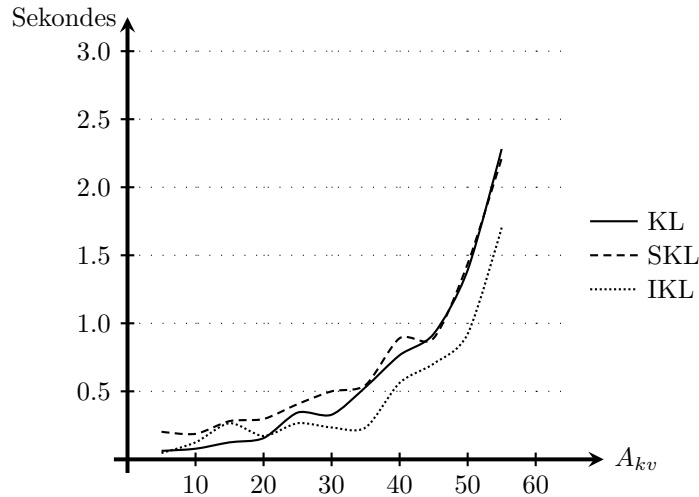
Die resultate van die prestasie van die verskillende algoritmes vir die oplossing van $K/E, A/G$ -HTP's word in die volgende afdelings verskaf vir verskillende situasies.

7.4.2.1 Verskillende verhoudings van konkawe en konvekse funksies

Die oplossingstye vir die onderskeie algoritmes wat in hierdie hoofstuk voorgestel word, word in Tabel 7-1 gegee. Hierdie oplossingstye is verkry deur elke keer 100 probleme lukraak te genereer wat A_{kv} konvekse en $60 - A_{kv}$ konkawe funksies in die doelfunksie bevat. 'n Grafiese voorstelling van hierdie data word in Figuur 7.3 verskaf. Uit die data kan die gevolgtrekking gemaak word dat die verbeterde vertak-en-begrens-algoritmes vir hierdie keuse van probleme beter vaar as die gewone vertak-en-begrens-algoritme. Die SKL-algoritme vaar marginaal beter as die KL-algoritme in terme van oplossingstyd. Dit is egter nie altyd die geval nie. Daar is baie $K/E, A/G$ -HTP's waarvoor die SKL-algoritme se oplossingstye stadiger is as die van die KL-algoritme. Die verbetering van die IKL-algoritme op die oplossingstye van die KL-algoritme is wel beter, maar nie so dramaties as wat die geval was vir die soortgelyke algoritmes in Hoofstuk 5 nie.

Die aantal subprobleme wat die KL-, SKL-, IKL-algoritmes onderskeidelik skep om 100 lukraak gegenereerde HTP's op te los, word in Tabel 7-2 gegee. Hierdie data is verkry deur elke keer 100 probleme lukraak te genereer wat A_{kv} konvekse en $60 - A_{kv}$ konkawe funksies in die doelfunksie bevat. 'n Grafiese voorstelling van hierdie data word in Figuur 7.4 verskaf. Die data vir die aantal subprobleme wat elke algoritme skep, is soortgelyk aan die verloop van die oplossingstye en soortgelyke gevolgtrekkings kan uit hierdie data gemaak word. Daar is nie 'n beduidende afname in die aantal subprobleme van die KL- na die SKL-algoritme nie. Die IKL-algoritme skep egter minder as die helfte van die aantal subprobleme wat die ander twee algoritmes skep.

'n Ander insiggewende maatstaf om na te kyk, is om die drie algoritmes relatief tot mekaar te meet. In Figuur 7.5 word die oplossingstye vir elk van die drie algoritmes relatief tot die KL-algoritme gegee. Dit is interessant om daarop te let dat die SKL- en



Figuur 7.3: Grafieke van die totale oplossingstyd in sekondes van 100 lukraak gegenereerde $K/E, A/G$ -HTP's wat elkeen 60 funksies in die doelfunksie het, waarvan A_{kv} konvekse funksies is. Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die volgende afkortings word gebruik: KL = vertak-en-begrens-algoritme vir $K/E, A/G$ -HTP's, SKL = vertak-sny-en-begrens-algoritme vir $K/E, A/G$ -HTP's, IKL = isogrens-vertak-en-begrens-algoritme vir $K/E, A/G$ -HTP's.

IKL-algoritmes toenemend beter vaar relatief tot die KL-algoritme met 'n toename in die aantal konvekse funksies in die doelfunksie.

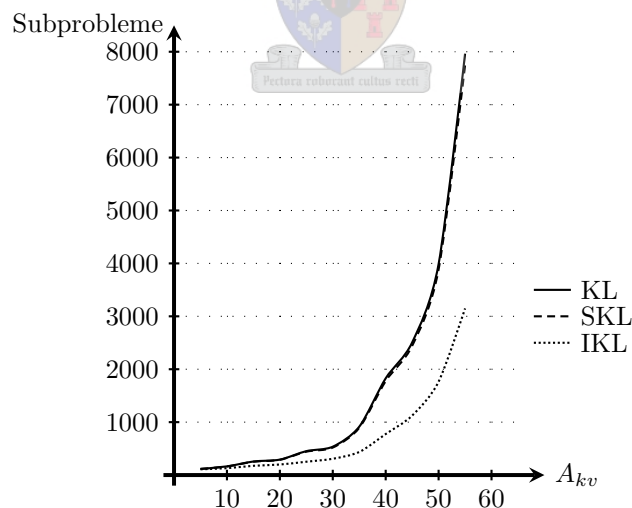
A_{kv}	Algoritme		
	KL	SKL	IKL
5	116	116	109
10	164	162	130
15	256	252	175
20	292	288	199
25	452	442	252
30	536	520	308
35	914	892	436
40	1828	1776	775
45	2506	2442	1124
50	3988	3886	1765
55	7964	7754	3149

Tabel 7-2: Die totale aantal subprobleme wat opgelos moet word om 100 lukraak gegenereerde $K/E, A/G$ -HTP's op te los wat elkeen 60 funksies in die doelfunksie het, waarvan A_{kv} konvekse funksies is. Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die volgende afkortings word gebruik: KL = vertak-en-begrens-algoritme vir $K/E, A/G$ -HTP's, SKL = vertak-sny-en-begrens-algoritme vir $K/E, A/G$ -HTP's, IKL = isogrens-vertak-en-begrens-algoritme vir $K/E, A/G$ -HTP's.

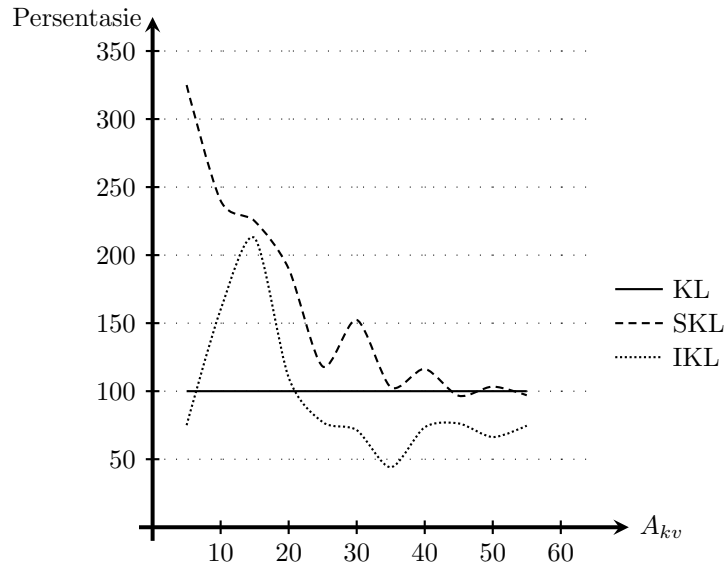
In Figuur 7.6 word die aantal subprobleme gegee wat elkeen van die drie algoritmes relatief tot die KL-algoritme skep om 100 $K/E, A/G$ -HTP's op te los. Die SKL- en IKL-algoritmes skep minder subprobleme as die KL-algoritme. Beide hierdie algoritmes vaar toenemend beter relatief tot die KL-algoritme met 'n toename in die aantal konvekse funksies in die doelfunksie.

7.4.2.2 Invloed van gemiddelde helling

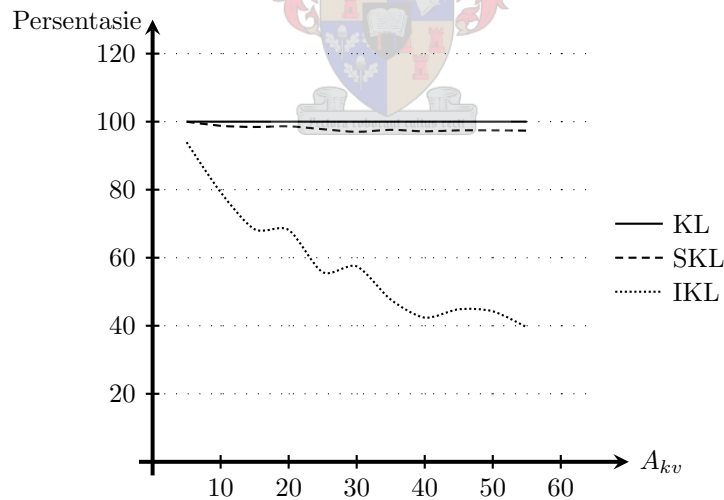
Die invloed van die intervallengte waaruit die gemiddelde hellings lukraak gegenereer word, word voorts ondersoek. Laat Ω die intervallengte wees waaruit die gemiddelde hellings gekies word. Dit is te wagte uit die resultate in Hoofstuk 5 dat die aantal subprobleme wat tydens die algoritmes gegenereer word, en dus die oplossingstye daarvan, sal toeneem met 'n afname in die intervallengte waaruit die gemiddelde hellings gegenereer word. 'n Eksperiment is gedoen deur lukraak probleme te genereer waarin die interval waaroor die gemiddelde hellings kan varieer, verstel kan word. In Tabel 7-3 word hierdie resultate getoon. Dieselfde resultate volg in Figuur 7.7. Geen resultate kon gegenereer word vir $\Omega < 1$ nie, omdat die aantal subprobleme wat die algoritmes skep in die oplossing van die probleem sodanig toeneem dat dit nie in die toegelate 64 KB geheue per subroetine gestoor kan word nie. Dit is duidelik dat die oplossingstye beduidend daal met 'n toename in Ω .



Figuur 7.4: Grafieke van die totale aantal subprobleme wat opgelos moet word om 100 lukraak gegenereerde $K/E, A/G$ -HTP's op te los wat elkeen 60 funksies in die doelfunksie het, waarvan A_{kv} konvekse funksies is. Die veranderlikes se bogrense is lukraak gegenereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegenereer. Die volgende afkortings word gebruik: KL = vertak-en-begrens-algoritme vir $K/E, A/G$ -HTP's, SKL = vertak-sny-en-begrens-algoritme vir $K/E, A/G$ -HTP's, IKL = isogrens-vertak-en-begrens-algoritme vir $K/E, A/G$ -HTP's.



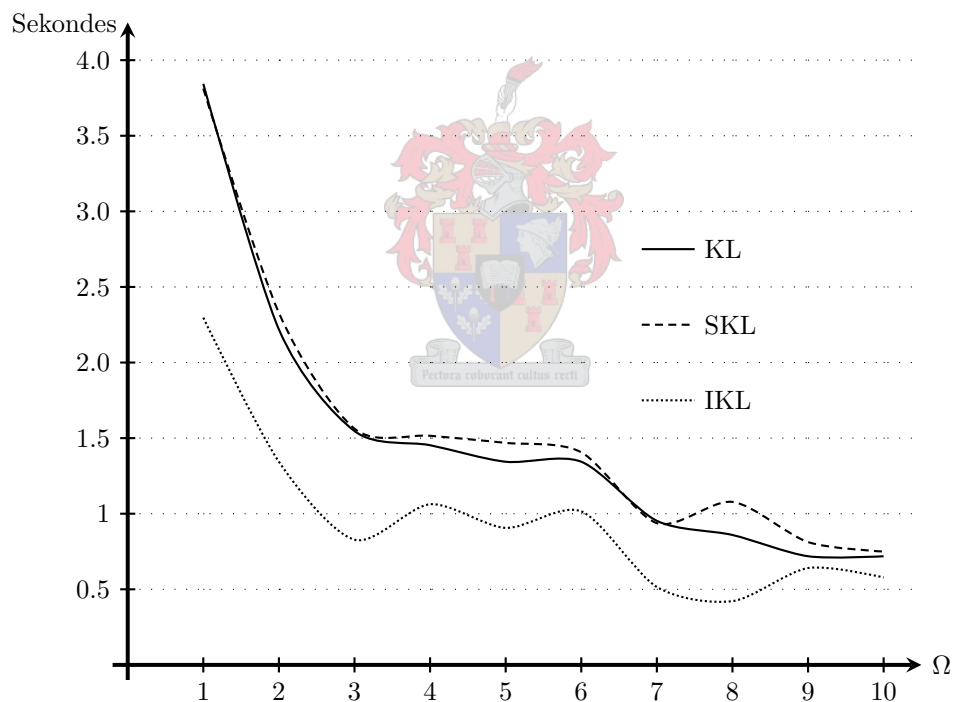
Figuur 7.5: Grafieke van die persentasie van die oplossingstyd in sekondes relatief tot die oplossingstyd van die normale vertak-en-begrens-algoritme om 100 lukraak gegengereerde $K/E, A/G$ -HTP's op te los wat elk 60 funksies in die doelfunksie het, waarvan A_{kv} konvekse funksies is. Die veranderlikes se bogrense is lukraak gegengereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegengereer.



Figuur 7.6: Grafieke van die persentasie subprobleme wat opgelos moet word relatief tot die aantal subprobleme van die normale vertak-en-begrens-algoritme om 100 lukraak gegengereerde $K/E, A/G$ -HTP's op te los wat elk 60 funksies in die doelfunksie het, waarvan A_{kv} konvekse funksies is. Die veranderlikes se bogrense is lukraak gegengereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is. Die gemiddelde hellings van die funksies is lukraak uit die interval $[5, 8]$ gegengereer.

Ω	Algoritme		
	KL	SKL	IKL
1	3.8438	3.8125	2.2969
2	2.2188	2.3281	1.3438
3	1.5469	1.5625	0.8281
4	1.4531	1.5156	1.0625
5	1.3438	1.4688	0.9063
6	1.3438	1.4063	1.0156
7	0.9531	0.9375	0.5156
8	0.8594	1.0781	0.4219
9	0.7188	0.8125	0.6406
10	0.7188	0.7500	0.5781

Tabel 7-3: Die totale oplossingstyd vir 100 lukraak gegengereerde $K/E, A/G$ -HTP's wat elk uit 60 funksies bestaan teenoor intervallengte Ω waaruit die gemiddelde hellings gegengereer is. 'n Mengsel van 55 konvekse en 5 konkawe funksies in die doelfunksie is gebruik. Die veranderlikes se bogrense is lukraak gegengereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is.



Figuur 7.7: Grafieke van die totale oplossingstye om 100 lukraak gegengereerde $K/E, A/G$ -HTP's wat elk uit 60 funksies bestaan teenoor intervallengte Ω waaruit die gemiddelde hellings gegengereer is. 'n Mengsel van 55 konvekse en 5 konkawe funksies in die doelfunksie is gebruik. Die veranderlikes se bogrense is lukraak gegengereer sodat die intervallengte van die veranderlikes tussen 100 en 200 eenhede is.

7.4.2.3 Invloed van konvekse tob-funksie

Uit eksperimente word dit gou duidelik dat die oplossingstyd toeneem soos wat die aantal konvekse funksies in die doelfunksie van die probleem toeneem. Dit wil ook uit ekspe-

rimente voorkom of probleme waarvan die optimale oplossing 'n konvekse funksie in die versameling \mathcal{K} , het 'n langer oplossingstyd het as die waarin daar geen konvekse funksies in \mathcal{K} is nie. Daar is gepoog om 'n versameling van subprobleme te genereer waarin verseker word dat daar in die optimale oplossing altyd een konvekse tob-funksie is. Dit is egter feitlik onmoontlik om dit reg te kry vir groter waardes van n . Daar is kode geskep om net die oplossingstye van oplossings waar daar wel 'n konvekse tob-funksie is te stoor en dan hierdie subversameling se oplossingstye te vergelyk met die oplossingstye van die gevalle waar daar geen konvekse tob-funksie in die optimale oplossing voorkom nie. Die geval waar daar 'n konvekse tob-funksie in die optimale oplossing voorkom, kom baie selde voor in die geval van lukraak gegenereerde probleme en die voorkoms daarvan neem af met 'n toename in die aantal funksies n . Indien die aantal funksies n minder as 10 funksies is, gebeur dit meer gereeld, maar dan is die oplossingstye so kort dat akkurate resultate onmoontlik is⁵.

Die volgende oplossingstye is verkry vir 100 $K/E, A/G$ -HTP's vir 55 konvekse en 5 konkawe funksies in die doelfunksie en waar die optimale oplossing een konvekse funksie in \mathcal{K} het: Die KL-algoritme neem 7.8437 sekondes, die SKL-algoritme neem 8.3437 sekondes en die IKL-algoritme neem 6.5937 sekondes om die 100 probleme op te los. Hierdie tye is beduidend (ongeveer drie maal) stadiger as die data wat in Tabel 7-1 gegee word.

7.4.3 Gevolgtrekkings

Daar bestaan geen spesialis algoritmes vir $K/E, A/G$ -HTP's nie, wat beteken daar is geen bestaande algoritmes waarteen die algoritmes wat hier voorgestel word gemeet kan word nie. Die KL-algoritme word as bakenalgoritme gekies omdat dit die veralgemening van die gewone vertak-en-begrens-algoritme is.

'n Isogrens-vertak-sny-en-begrens-algoritme is ook geprogrammeer, maar die oplossingstye daarvan is min of meer dieselfde (indien nie swakker nie) as die gewone isogrens-vertak-en-begrens-algoritme. Omdat hierdie algoritme nie 'n wesentlike verbetering op oplossingstyd teweeg bring nie, word dit nie hier aangebied nie.

Alhoewel die algoritmes wat hier aangebied word, verbeter op die gewone vertak-en-begrens-algoritme, is die verbetering nie so drasties as wat die geval was vir die $K/E/G$ -HTP's nie.

⁵ VBA gebruik die sisteemtyd van die rekenaar om die oplossingstye te bereken. Die kleinste tydsinterval wat die rekenaar kan registreer, is 'n honderdste van 'n sekonde. Die tipiese oplossingstyd vir so 'n klein $K/E, A/G$ -HTP is kleiner as hierdie waarde, wat die tydsberekening onakkuraat maak.



Hoofstuk 8

Slotopmerkings

In hierdie hoofstuk word 'n kort opsomming gegee van wat in die proefskrif bereik is; 'n oorsigtelike samevatting word verskaf; idees vir verdere studie word gegee en ter afsluiting volg 'n bespreking oor hoe en waar die doelwitte wat in Hoofstuk 1 vir hierdie proefskrif gestel was, bereik is.

8.1 Proefskrifopsomming

In hierdie proefskrif word (hoofsaaklik) vertak-en-begrens-metodes aangebied vir die oplossing van nielineêre HTP's.

In die eerste hoofstuk van hierdie proefskrif is 'n kort beskrywing gegee van die probleem wat in die proefskrif ondersoek sou word. Daar is ook 'n nuwe notasie voorgestel om na verskillende tipes HTP's te verwys en 'n aantal toepassings van HTP's is verskaf.

In Hoofstuk 2 is bestaande literatuur kortliks bespreek. Slegs vier publikasies kon opgespoor word waarin spesifiek $K/E/G$ -HTP's (die getransformeerde ekwivalent van $N/A/G$ -HTP's) beskou word [9, 27, 36, 41]. Twee van hierdie vier publikasies verskaf net heuristiese benaderings [27, 41]. Die ander twee artikels [9, 36] bevat wel eksakte metodes, maar hierdie outeurs beskou slegs kwadratiese funksies in die doelfunksie en heeltallige oplossings onderskeidelik.

'n Kort oorsig oor bestaande teoretiese resultate is ook in Hoofstuk 2 gegee. Dit is hoofsaaklik minder bekende teorie oor nielineêre optimering. In hierdie hoofstuk is klassieke nodige en voldoende voorwaardes beskou vir nielineêre optimering.

In Hoofstuk 3 is $K/E/V$ -HTP's beskou. Twee nuwe algoritmes is aangebied wat hierdie subversameling van HTP's kan optimeer. Beide hierdie algoritmes, naamlik die breuk- en die hellingalgoritme, gebruik die feit dat die funksie wat 'n toedeling tussen sy bo- en ondergrens kry (waarvan daar hoogstens een kan wees) altyd 'n vaste toedeling sal kry. Beide hierdie algoritmes lewer beduidend beter oplossingstye as bestaande vertak-en-begrens-metodes.

In Hoofstuk 4 is soekalgoritmes aangebied vir die oplossing van $K/E/G$ -HTP's. Die

blaarsoek- en taksoekalgoritmes is voorgestel. Beide gebruik 'n binêre soektog om die optimale oplossing te bepaal. Hierdie algoritmes lewer egter swak resultate indien dit met bestaande metodes vergelyk word.

'n Heuristiek en drie algoritmes, naamlik die isogrens-vertak-en-begrens-algoritme, vertak-sny-en-begrens-algoritme en isogrens-vertak-sny-en-begrens-algoritme, is in Hoofstuk 5 aangebied. Die heuristiek kan onder die regte omstandighede die optimale oplossing van 'n HTP waarborg. Die drie algoritmes, maar veral die eerste en die laaste een, lewer besonder goeie resultate en bring 'n groot besparing aan oplossingstyd mee relatief tot die bestaande metodes om hierdie probleem op te los.

In Hoofstuk 6 is ondersoek ingestel na 'n kontinue, in plaas van 'n vertak-en-begrens-benadering. Die eenvoudiger geval van slegs een konvekse en een konkawe funksie in die doelfunksie is eerstens beskou. Die resultate vir hierdie geval is dan veralgemeen na meer konkawe en konvekse funksies in die doelfunksie. Hierdie benadering blyk egter nie effektief te wees nie.

'n Vertak-en-begrens-benadering is in Hoofstuk 7 gevolg vir die oplossing van K/E , A/G -HTP's. Dieselfde algoritmes wat in Hoofstuk 5 vir die goeie resultate verantwoordelik was, is veralgemeen om die versameling konkawe funksies ook te akkommodeer. Die isogrens-vertak-en-begrens-algoritme lewer wel beter oplossings as gewone vertak-en-begrens-algoritmes, maar ongelukkig is die verbetering in oplossingstyd in hierdie geval nie so drasties as wat dit in Hoofstuk 5 was nie. Dit is interessant dat die benadering om die dele van die toelaatbare gebied waarin die optimale oplossing reeds bekend is, in die vertakkingsproses af te sny, nie in hierdie geval 'n beduidende verbetering in oplossingstyd kan bewerkstellig nie. In sommige gevalle neem die berekening van die intervalle wat afgesny kan word langer as die besparing in die aantal subprobleme wat opgelos moet word.



8.2 Samevatting en gevolgtrekking

Aan die einde van elke hoofstuk is 'n samevatting met gevolgtrekkings vir daardie hoofstuk gegee. Hier volg dus slegs 'n paar oorsigtelike samevattende idees.

Aan die begin van die studie is 'n besluit geneem om ondersoek in te stel na algoritmes wat 'n definitiewe optimale oplossing waarborg. Heuristiese benaderings word dus op een uitsondering na nêrens in hierdie studie ondersoek nie.

Na 'n vergelyking van heelwat verskillende benaderings (soos dinamiese programmering, soektogte deur oplossings wat aan die nodige voorwaardes voldoen en lineêre omsluitings waarin daar tydens elke iterasie 'n ekstra lineêre beperking bygevoeg word), het dit duidelik geword dat vertak-en-begrens-algoritmes die effektiefste is en daarom is daar op hierdie tipe algoritmes gekonsentreer. Deurgaans is die gewone (standaard) vertak-en-begrens-metode dus as 'n metingspunt (Eng.: *benchmark*) gebruik, siende dat hierdie metode sonder uitsondering beter resultate gelever het as die res van die standaard metodes.

Waar daar slegs konvekse funksies in die doelfunksie voorkom, veroorsaak die nodige voorwaardes direk dat die probleem gediskretiseer word, wat die vertak-en-begrens-metodes

bevoordeel omdat dit juis diskrete dele uit die toelaatbare gebied kan sny.

In die geval waar daar konvekse en konkawe funksies in die doelfunksie voorkom, is die oplossings kontinu en verstadig dit die konvergensie van die vertak-en-begrens-benadering. In hierdie geval moet die takke as't ware "inzoom" op die oplossing.

Gedurende die ondersoek is 'n groot hoeveelheid tyd en energie daaraan bestee om kontinue metodes (dit wil sê metodes wat nie van vertak-en-begrensing gebruik maak nie) te ondersoek. Hierdie proefskrif weerspieël nie werklik hierdie ondersoeke nie, hoofsaaklik omdat geen effektiewe metode opgespoor kon word nie. Hoofstuk 6 raak aan hierdie onderwerp en gee 'n kort blik op van die rigtings wat ingeslaan is. Die uiteindelijke gevolgtrekking na al hierdie pogings is egter dat 'n kontinue benadering nie kompeterende metodes kan daarstel wat met die vertak-en-begrens-algoritmes kan meeding nie.

8.3 Validering van algoritmes

Regdeur die proefskrif is al die algoritmes wat geprogrammeer is, getoets en gevalideer. Hierdie validering is gedoen deur groot hoeveelhede probleme lukraak te skep en op te los en dan die oplossings met mekaar te vergelyk.

As eerste stap is die dinamiese-programmeringsalgoritme getoets teen LINGO [19]. 'n Duisend van elke tipe probleem (dit is $K/E/V$ -, $K/E/G$ -, $K/E, A/G$ -HTP's) is lukraak geskep en oplos met dinamiese programmering en LINGO waarin getoets word dat hulle presies dieselfde oplossings lewer. In die LINGO-formulering is gebruik gemaak van stuks-gewyse liniëring van die funksies in die doelfunksie. In die geval van $K/E, A/G$ -HTP's gee die twee metodes nie presies dieselfde oplossings nie omdat dinamiese programmering slegs die beste heeltallige oplossings lewer. Hierna is dieselfde toetse met die gewone vertak-en-begrens-algoritme en LINGO gedoen, waarna die eksperiment herhaal is deur die oplossings van die gewone vertak-en-begrens-algoritmes met dinamiese programmering te vergelyk. Drieduisend probleme (per tipe) is dus lukraak gegenereer en opgelos waarin die drie benaderings dieselfde oplossing gegee het.

Om die ander algoritmes te toets, is die validerings van nuwe algoritmes teenoor dinamiese programmering en die gewone vertak-en-begrens-algoritme gedoen. Dit is ook gedoen deur elke keer duisend probleme lukraak te genereer, op te los en die oplossings te vergelyk.

As 'n verdere toets, bo en behalwe die duisend toetslopies, vir korrektheid van die algoritmes is al die algoritmes wat geskep is om 'n spesifieke tipe probleem op te los se oplossings met mekaar vergelyk tydens die generering van die resultate. Dit beteken dat daar in sommige gevalle tot 1200 probleme lukraak gegenereer is, met twee of drie verskillende algoritmes opgelos is en al hierdie oplossings dan met mekaar vergelyk is om seker te maak dat al die algoritmes in al die gevalle dieselfde oplossings gee.

'n Ander manier van validering wat ook gebruik is, is om 'n minder algemene probleem met algoritmes wat 'n meer algemene probleem kan oplos, op te los. Twee voorbeelde hiervan is om (1) $K/E/V$ -HTP's met die algoritmes vir $K/E/G$ -HTP's op te los en (2) $K/E/G$ - en $K/E/V$ -HTP's met die algoritmes vir $K/E, A/G$ -HTP's op te los. Op hierdie

manier kon feitlik alle algoritmes vir minstens spesiale gevalle met mekaar vergelyk word.

8.4 Idees vir verdere studie

Die soeke na oplossings lei feitlik altyd tot nuwe vrae en die oplossings daarvoor skep weer op hul beurt nuwe vrae, ensovoorts. In enige studie, en dus ook in hierdie proefskrif, kom daar dus 'n punt waar die ondersoek afgesny moet word, of nuwe vrae kan *ad infinitum* bygevoeg word. In hierdie afdeling word van hierdie nuwe vrae genoem, wat dan op hul beurt idees oplewer oor hoe hierdie ondersoek uitgebrei en verdiep kan word.

8.4.1 Idees vir die uitbreiding van hierdie studie

Die probleem wat in hierdie proefskrif aangepak is, kan veralgemeen word na 'n HTP met konvekse, konkawe en S-krommes in die doelfunksie en met ekstra beperkings (bo en behalwe die grense op die x_i 's). Hierdie ekstra beperkings kan byvoorbeeld wees dat een projek 'n toedeling tot bo 'n sekere vlak moet kry voordat 'n volgende projek 'n toedeling mag ontvang. Hierdie probleem kan as 'n begrotingsprobleem beskou word en in die algemeen word dit dan 'n knapsakprobleem met ekstra beperkings, naamlik

$$\text{maksimeer } \sum_{i=1}^n w_i f_i(x_i) \quad (8.1)$$

onderhewig aan

$$\sum_{i=1}^n x_i \leq g(b) \quad (8.2)$$

$$l_i \leq x_i \leq u_i \quad i = 1, 2, \dots, n \quad (8.3)$$

$$-(x_l - d_l) \leq My \quad (8.4)$$

$$x_j - t_j \leq M(1 - y) \quad (8.5)$$

$$y \quad 0, 1 \text{ veranderlike}$$

waar w_i die gewig van die i -de projek in die begroting is, $f_i(x_i)$ die opbrengs van projek i by befondsingsvlak x_i en $g(b)$ die bedrag is waarmee begroot kan word. Hierdie bedrag kan konstant wees of varieer (die sogenaamde knapsak-wat-rek (of -krimp)-probleem). Gevalle waar hierdie situasie kan ontstaan, is indien groter (en/of verdere) bedrae toegestaan word aan afdelings of projekte wat beter as 'n sekere vlak presteer. Die begrotingsbedrag is immers nie in yster gegiet nie. Beperkings (8.4) en (8.5) verseker dat, indien projek j bo 'n vlak t_j befonds word, projek l tot minstens vlak d_l befonds moet word. M is natuurlik 'n groot getal. Behalwe vir gewone vertak-en-begrensing bestaan daar nie oplossingsmetodes vir hierdie veralgemeende probleem nie en lê hier dus nog 'n braak veld vir verdere studie.

Hierdie meer algemene formulering gee dus aanleiding tot 'n knapsakprobleem met

1. 'n nielineêre doelfunksie in vergelyking (8.1);
2. 'n konstante of veranderbare begrotingsbedrag in vergelyking (8.2);
3. bo- en ondergrense op die befondsingsvlakke van elke projek in vergelyking (8.3); en
4. voorwaardes op die befondsingsvlakke van enkele van die projekte. Hierdie voorwaardes kan die volgende vorm aanneem: indien 'n sekere projek i hoër as 'n sekere vlak (sê $x_i > d_i$) befonds moet word, moet 'n ander projek, sê j , tot op 'n gegewe (toetree-) vlak (t_j) befonds word. Dit wil sê indien projek i hoër as d_i befonds word, moet projek j tot minstens by t_j befonds word (vergelykings (8.4) en (8.5)).

Die probleem waar daar slegs konvekse funksies (en later konvekse en konkawe funksies) in die doelfunksie is en daar slegs onder- en bogrense op die veranderlikes x_i is, het so 'n enorme uitdaging gebied dat hierdie proefskrif slegs hieroor handel.

Die oplossing van hierdie probleem waar daar ook S -krommes voorkom van die vorm

$$f_i(x_i) = \frac{k_i}{1 + e^{-(s_i + m_i x_i)}} \quad (8.6)$$

$$f_i(x_i) = e^{s_i - \frac{m_i}{x_i}} \quad (8.7)$$

$$f_i(x_i) = a_i s_i^{m_i x_i} \quad a_i > 0; 0 < s_i < 1; m_i < 1 \quad (8.8)$$

$$f_i(x_i) = a_i + (k_i - a_i) \frac{x_i^{s_i}}{m_i^{s_i} + x_i^{s_i}}, \quad s_i > 1, \quad (8.9)$$

waar s_i , m_i en c_i , met $i = 1, 2, 3 \dots, n$ parameters is waarmee die vorm van die funksie gestel kan word en k_i en a_i konstantes in die doelfunksie is, voorkom bly dus steeds 'n ope vraag. Die versameling van S -krommes in (8.6) tot (8.9) word in Lilien en Kotler [18] gebruik.

Die geval waar die begroting varieerbaar is, is ook glad nie ondersoek nie en kan in toekomstige studies aangepak word.

Die geval van 'n knapsak met ekstra beperkings, oftewel die geval waar voorwaardes op die befondsingsvlakke van sekere projekte geplaas word indien/voor ander projekte befonds word, is nog oop vir ondersoek.

Ondersoeke waarin spesiale gevalle beskou word, soos byvoorbeeld die geval waar al die funksies in die doelfunksie kwadratiese funksies is, kan dalk meer lig op die gedrag, oplossings en werking van algoritmes werp.

8.4.2 Idees vir verdieping van sekere aspekte in hierdie studie

Die geval van die herleiding na een konvekse en een konkawe funksie kan verder ondersoek word. Spesifiek kan aandag gegee word aan aspekte van die uniekheid van en/of aantal punte waarby die hellings gelyk aan mekaar kan wees soos wat in die nodige voorwaardes vereis word. 'n Ondersoek na die spesiale geval waar hierdie funksies almal oor

dieselfde intervale strek, kan dalk ook waardevolle insig oor die oplossing van funksies oor wisselende intervale verskaf.

Die vertak-en-begrens-algoritmes, sowel as die vertak-sny-en-begrens-algoritmes, kan verder verfyn word deur ekstra toetse na die oplos van elke subprobleem by te voeg. Een so 'n toets kan byvoorbeeld wees om te toets of ekstra beperkings wat as gevolg van die vertakkings bygevoeg is, nie dalk meer as een veranderlike begrens om 'n tob-funksie te wees nie. Uit die nodige voorwaardes is dit duidelik dat daar hoogstens een tob-funksie kan wees, wat impliseer dat, indien die ekstra beperking twee funksies na tob-funksies forseer, die optimale oplossing nie in daardie tak kan voorkom nie.

In die ondersoek na die $K/E, A/G$ -HTP moet 'n hele aantal subprobleme opgelos word om die regte toedeling vir die konvekse tob-funksies te bepaal. Dit is bekend vanuit die nodige voorwaardes dat die hellings van alle tob-funksie dieselfde moet wees. Verdere ondersoek kan ingestel word na moontlikhede om die feit te gebruik om die konvergensie te versnel.

Uit 'n suiwer programmeringsoogpunt kan die kode vir die algoritmes verdere aandag kry. So 'n ondersoek sal egter meer vanuit 'n rekenaarwetenskapfokus (en kompleksiteitsteorie) as uit die oogpunt van operasionele navorsing moet geskied. Meer spesifiek kan die volgende ondersoek word:

1. Die kode vir die implementering van die algoritmes kan verder ondersoek word om dit meer effektief (vinniger) te maak. Die volgorde en/of patroon waarin die subprobleme opgelos word, kan miskien die algoritmes aansienlik vinniger maak.
2. Die kode vir die oplossing van die algoritmes kan uitgebrei word om meer funksies te hanteer. Weens 'n gebrek aan geheue per subroetine in VBA kon daar nie werklike groot probleme aangepak word nie. Die ideaal sou wees om kode te skep wat 1 500 en meer funksies kan hanteer.

8.5 Bereiking van doelwitte

In Hoofstuk 1 word die volgende doelwitte gestel:

1. Verskaf 'n oorsig oor watter werk en metodes in die literatuur oor hierdie onderwerp bestaan.
2. Gee 'n oorsig oor die bestaande teorie wat as fondament van die oplossing van nielineêre HTP's dien.
3. Ontwikkel algoritmes om die HTP waarvan die doelfunksie uit skeibare, monotoon stygende konvekse funksies bestaan, op te los. Ondersoek die oplossingstyd van hierdie algoritmes.
 - (a) Ontwikkel eers algoritmes vir die spesiale (eenvoudiger) geval waar al die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$ oor vaste intervale geoptimeer word.

- (b) Ontwikkel dan algoritmes vir die meer algemene geval waar die funksies $f_i(x_i)$ met $i = 1, 2, \dots, n$ oor wisselende intervalle geoptimeer word.
4. Ontwikkel algoritmes om die HTP waarvan die doelfunksie uit skeibare, monotoon stygende konvekse en konkawe funksies bestaan, op te los. Onderzoek die oplossingstyd van hierdie algoritmes.
 5. Vergelyk verskillende algoritmes met mekaar en maak aanbevelings oor die spoed waarmee dit probleme kan oplos en dus die bruikbaarheid daarvan.
 6. Verskaf 'n kort samevatting oor wat bereik is en gee idees oor moontlikhede vir verdere studie.

Die eerste doelwit word in §2.1 aangespreek. Uit hierdie afdeling is dit duidelik dat daar bitter min navorsing oor $K/E/G$ -HTP's gedoen is. Die werk wat wel gepubliseer is, konsentreer hoofsaaklik op heuristiese oplossingsmetodes. Die res van Hoofstuk 2 handel oor die tweede doelwit.

Doelwit 3a word in Hoofstuk 3 ondersoek. Hier word twee nuwe algoritmes voorgestel wat in wese neerkom op 'n soektog deur al die oplossings wat aan die nodige voorwaardes van optimaliteit voldoen. Uit die resultate en samevatting van hierdie hoofstuk volg dat beide hierdie twee algoritmes goeie resultate lewer relatief tot bestaande tegnieke; naamlik dinamiese programmering en die gewone vertak-en-begrens-algoritme. Dit word ook uit hierdie resultate duidelik dat dinamiese programmering nie 'n goeie maatstaf van vergelyking is nie, hoofsaaklik omdat dinamiese programmering se kompleksiteit toeneem met 'n toename in die intervalle van die veranderlikes x_i .

Die doelwit wat in punt 3b gestel word, geniet aandag in Hoofstukke 4 en 5. In Hoofstuk 4 word gepoog om die soekalgoritmes van Hoofstuk 3 te veralgemeen, maar uit die resultate kan afgelei word dat hierdie rigting van ondersoek vrugteloos is. In Hoofstuk 4 word gekyk na aansienlik beter metodes om die $K/E/G$ -HTP aan te pak. Al die algoritmes in hierdie hoofstuk fokus daarop om eerder die lineêre omsluiting van die probleem op te los en dan van een of ander vorm van vertak-en-begrensing en/of snitte uit die toelaatbare gebied gebruik te maak. Hierdie algoritmes lewer uitstekende resultate. Een van die groot deurbrake is dat 'n liniëring van die betrokke nielineêre probleem opgelos kan word sonder om ooit een simpleksiterasie te doen.

Doelwit 4 is die fokus van Hoofstukke 6 en 7.

Die voorlaaste doelwit word aan die einde van elke hoofstuk aangespreek in die afdelings "resultate" en "samevatting".

Die heel laaste doelwit word in die afdelings hierbo aangespreek.

Die hoofbydrae van hierdie proefskrif is vervat in Hoofstukke 4 tot 7.



Verwysings

- [1] BILLIONETT A, 2002, *Approximation algorithms for fractional knapsack problems*, Operations Research Letters, **30**, pp. 336–342.
- [2] BITRAN GR, 1989, *Tradeoff curves, targeting and balancing in manufacturing queuing networks*, Operations Research, **37**, pp. 547–564.
- [3] BITRAN GR & HAX AC, 1979, *On the solution of convex knapsack problems with bounded variables*, In: Survey of Mathematical Programming, ED. Prékoba A, pp. 357–364.
- [4] BITRAN GR & HAX AC, 1981, *Disaggregation and resource allocation using convex knapsack problems with bounded variables*, Management Science, pp. 431–441.
- [5] BRETTHAUER KM & SHETTY B, 1995, *The nonlinear resource allocation problem*, Operations Research, **43(4)**, pp. 670–683.
- [6] BRETTHAUER KM & SHETTY B, 2002, *A pegging algorithm for the nonlinear resource allocation problem*, Computers & Operations Research, **29**, pp. 505–527.
- [7] BRETTHAUER KM & SHETTY B, 2002, *The nonlinear knapsack problem: algorithms and applications*, European Journal of Operational Research, **138**, pp. 459–472.
- [8] CZUCHRA W, 1998, *Optimizing budget spendings for software implementation and testing*, Computers and Operations Research, **26**, pp. 731-747.
- [9] DAI Y-H & FLETCHER R, 2006, *New algorithms for singly linearly constraint quadratic programs subject to lower and upper bounds*, Mathematical Programming, **106**, pp. 403–521.
- [10] DE KOCK HC, 1988, *Ontwikkeling en toepassing van wiskundige programmeringstechnieke vir besluitneming in voermengselprobleme*, PhD-proefskrif, Universiteit van Stellenbosch, Stellenbosch.
- [11] DUDEWICZ EJ & DALAL SR, 1975, *Allocation of observations in ranking and selection with unequal variances*, [Aanlyn], [Laaste besoek Julie 2004], Beskikbaar by: <http://www.bilkent.edu.tr/~goren/simulation.html>

- [12] GERLA M & KLEINROCK L, 1977, *On the topological design of distributed computer networks*, IEEE Transactions on Communications, **25**, pp. 48–60.
- [13] HILLER FS & LIEBERMAN GJ, 2001, *Introduction to operations research*, 7^{de} uitgawe, McGraw-Hill, New York (NY).
- [14] HOCHBAUM DS, 1995, *A nonlinear knapsack problem*, Operations Research Letters, **17**, pp. 103–110.
- [15] HORST R & TUY H, 1995, *Global optimization: Deterministic approaches*, Springer, Berlyn.
- [16] IBARAKI T & KATO N, 1988, *Resource allocation problems*, MIT Press, Cambridge (MA).
- [17] KASANA HS & KUMAR KD, 2004, *Introductory Operations Research: Theory and applications*, Springer-Verlag, Heidelberg.
- [18] LILIE GL & KOTLER P, 1983, *Marketing Decision Making: A Model-building approach*, Harper & Row Publishers, New York (NY).
- [19] LINGO, 2006, *Lindo systems' index page*, [Aanlyn], [Laaste besoek Augustus 2006], Beksikbaar by <http://www.lindo.com/>
- [20] LOOTSMA FA, MENSCH TCA & VOS FA, 1990, *Multi-criteria analysis and budget reallocation in long-term research planning*, European Journal of Operational Research, **47(3)**, pp. 293–305.
- [21] LUENBERGER DG, 1973, *Introduction to linear and nonlinear programming*, Addison-Wesley Publishing Company Inc., Reading (MA).
- [22] LUSS H & GUPTA TCA, 1975, *Allocation of effort resources among competing activities*, Operations Research, **23(2)**, pp. 360–366.
- [23] MALONEY BM & KLEIN CM, 1993, *Constrained multi-item inventory systems: an implicit approach*, Computers and Operations Research, **20**, pp. 639–649.
- [24] MARTELLO S & TOTH P, 1990, *Knapsack problems: Algorithms and computer implementations*, John Wiley & Sons, Chichester, Engeland.
- [25] MCCORMICK GP, 1983, *Nonlinear programming: Theory, Algorithms and Applications*, John Wiley & Sons, New York (NY).
- [26] MELMAN A & RABINOWITZ G, 2000, *An efficient method for a class of continuous nonlinear knapsack problems*, Industrial Applied Mathematics, **42(3)**, pp. 440–448.
- [27] MORÉ JJ & VAVASIS SA, 1991, *On the solution of concave knapsack problems*, Mathematical Programming, **49**, pp. 397–411.

- [28] NAHAY JM, 2004, *Differential resolvents are complete and useful*, [Aanlyn], [Laaste besoek Mei 2006], Beskikbaar by <http://math.hunter.cuny.edu/ksda/papers/nahay.pdf>
- [29] PATRIKSSON M, 2005, *A Survey on a Classic Core Problem In Operations Research*, Verslag, Departement Wiskunde, Chalmers Universiteit van Tegnologie, Gteborg (Gothenburg), Swede, Aanlyn beskikbaar by: <http://www.cs.chalmers.se/~mipat/nlp.html>
- [30] PATRIKSSON M, 2006, *E-poskommunikasie*, Departement Wiskunde, Chalmers Universiteit van Tegnologie, Gteborg (Gothenburg), Swede, e-posadres: mipat@math.chalmers.se
- [31] PATRIKSSON M, 2006, *A survey on continuous nonlinear resource allocation problems*, Departement Wiskunde, Chalmers Universiteit van Tegnologie, Gteborg (Gothenburg), Swede, Aanlyn beskikbaar by: http://www.cs.chalmers.se/~mipat/LATEX/survey_0602.pdf, Artikel ingedien vir publikasie.
- [32] PFERSCHY U, PISINGER D & WOEGINGER GJ, 1996, *Simple but efficient approaches for the collapsing knapsack problem*, *Discrete Applied Mathematics*, **77**, pp. 271–280.
- [33] RIHA WO & WALKER J, 1997, *An efficient algorithm for the Lagrangean dual of nonlinear knapsack problems with additional nested constraints*, *Journal of Computational and Applied Mathematics*, **78**, pp. 9–18.
- [34] SINCLAIR M, 1989, *Optimering*, Klasnotas, Departement Toegepaste Wiskunde, Universiteit van Stellenbosch.
- [35] SWOKOVSKI EW, 1990, *Caculus*, 5^{de} uitgawe, PWS-Kent Publishing Company, Boston.
- [36] SUN XL, WANG FL & LI D, 2005, *Exact algorithm for concave knapsack problems: Linear underestimation and partition method*, *Journal of Global Optimization*, **33**, pp. 15–30.
- [37] TAHA HA, 1987, *Operations research: An introduction*, 4^{de} uitgawe, MacMillan Publishing Company, New York (NY).
- [38] TANGIAN A, 2004, *Redistribution of university budgets with respect to the status quo*, *European Journal of Operational Research*, **157(2)**, pp. 409–428.
- [39] VAVASIS SA, 1992, *Local minima for indefinite quadratic knapsack problems*, *Mathematical Programming*, **54**, pp. 127–153.
- [40] VENTER G, 1994, *'n Doeltreffende algoritme vir die algehele minimum van die konkawe knapsakprobleem*, MSc-tesis, Universiteit van Suid-Afrika, Pretoria.

- [41] VENTER G & WOLVAARDT JS, 2005, *'n Algoritme vir die minimum van die konkawe knapsakprobleem*, ORiON, **21(1)**, pp. 13–31.
- [42] VISAGIE SE, 1997, *Wisselboustelsels in kleingraangebiede*, MSc-tesis, Universiteit van Stellenbosch, Stellenbosch.
- [43] WAGNER HM, 1958, *The dual simplex algorithm for bounded variables*, Naval Research Logistics Quarterly, **5(3)**, pp. 257–261.
- [44] WINSTON WL, 2003, *Operations Research Applications and Algorithms*, 4^{de} uitgawe, Duxbury Press, Kalifornië.
- [45] WOLVAARDT JS, 2006, *persoonlike kommunikasie*, Departement Besluitkunde, Universiteit van Suid-Afrika, Pretoria.



Bylae A

Kuhn-Tucker-voorwaardes

Die standaard formulering van 'n nielineêre programmeringsprobleem, soos verskaf in Luenberger [21] en Sinclair [34], word gegee deur

$$\text{minimeer } f(\underline{x})$$

onderhewig aan

$$\begin{aligned} h_j(\underline{x}) &= 0 & j &= 1, 2, \dots, m \\ g_k(\underline{x}) &\leq 0 & k &= 1, 2, \dots, p \\ \underline{x} &\in A \subseteq E^n, \end{aligned} \tag{A.1}$$

waar $m \leq n$ en al die funksies f, h_1, h_2, \dots, h_m , en $g_1, g_2, \dots, g_p \in C^2$.

Die Kuhn-Tucker voorwaardes vir (A.1) word in beide Luenberger [21] en Sinclair [34] met behulp van Stelling A.1.1 gegee.

Stelling A.1.1 (Kuhn-Tucker-voorwaardes)

Gestel \underline{x}^* is 'n relatiewe minimumpunt vir (A.1) en 'n reguliere punt. Dan bestaan daar vektore $\underline{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_m)$ en $\underline{\mu} = (\mu_1, \mu_2, \dots, \mu_p)$ met $\mu_k \geq 0$ vir alle $k = 1, 2, \dots, p$, sodat

$$\frac{\partial f(\underline{x}^*)}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{\partial h_j(\underline{x}^*)}{\partial x_i} + \sum_{k=1}^p \mu_k \frac{\partial g_k(\underline{x}^*)}{\partial x_i} = 0 \quad i = 1, 2, \dots, n \tag{A.2}$$

$$\mu_k g_k(\underline{x}^*) = 0 \quad k = 1, 2, \dots, p. \tag{A.3}$$

Die algemene formulering vir die probleem wat in hierdie proefskrif aangepak word, is

$$\text{maksimeer } f(\underline{x})$$

onderhewig aan

$$\begin{aligned} h_j(\underline{x}) &= 0 & j &= 1, 2, \dots, m \\ g_k(\underline{x}) &\geq 0 & k &= 1, 2, \dots, p \\ \underline{x} &\in A \subseteq E^n. \end{aligned} \tag{A.4}$$

waar $m \leq n$ en al die funksies f, h_1, h_2, \dots, h_m , en $g_1, g_2, \dots, g_p \in C^2$.

Die formulering in (A.4) kan herskryf word om met die standaardvorm van Luenberger [21] en Sinclair [34] ooreen te stem. Na hierdie aanpassing volg die formulering as

$$\text{minimeer } -f(\underline{x})$$

onderhewig aan

$$\begin{aligned} -h_j(\underline{x}) &= 0 & j &= 1, 2, \dots, m \\ -g_k(\underline{x}) &\leq 0 & k &= 1, 2, \dots, p \\ \underline{x} &\in A \subseteq E^n. \end{aligned} \quad (\text{A.5})$$

Die Kuhn-Tucker-voorwaardes volgens Luenberger [21] en Sinclair [34] word dan gegee deur

$$\frac{-\partial f(\underline{x}^*)}{\partial x_i} + \sum_{j=1}^m \lambda_j \frac{-\partial h_j(\underline{x}^*)}{\partial x_i} + \sum_{k=1}^p \mu_k \frac{-\partial g_k(\underline{x}^*)}{\partial x_i} = 0 \quad i = 1, 2, \dots, n \quad (\text{A.6})$$

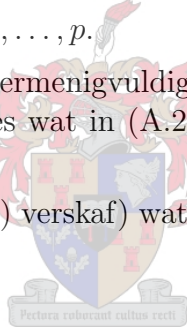
$$\mu_k \{-g_k(\underline{x}^*)\} = 0 \quad k = 1, 2, \dots, p. \quad (\text{A.7})$$

met $\underline{\lambda} \in E^n$ en $\mu_k \geq 0$ vir alle $k = 1, 2, \dots, p$.

Indien (A.6) en (A.7) elkeen met -1 vermenigvuldig word, is dit presies dieselfde formulering as die Kuhn-Tucker-voorwaardes wat in (A.2) en (A.3) gegee word. Hieruit volg die formulering van Stelling 2.2.

Die algemene formulering (soos in (1.1) verskaf) wat in hierdie proefskrif beskou word, is

$$\text{maks } \sum_{i=1}^n f_i(x_i)$$



onderhewig aan

$$\begin{aligned} \sum_{i=1}^n x_i &= B \\ x_i - l_i &\geq 0 & i &= 1, 2, \dots, n \\ u_i - x_i &\geq 0 & i &= 1, 2, \dots, n. \end{aligned}$$

en $l_i \geq 0$. Die Kuhn-Tucker-voorwaardes hiervoor volg dus as

$$\begin{aligned} f'_i(x_i) + \lambda + v_i - w_i &= 0 & i &= 1, 2, \dots, n \\ v_i(x_i - l_i) &= 0 & i &= 1, 2, \dots, n \\ w_i(-x_i + u_i) &= 0 & i &= 1, 2, \dots, n \\ \lambda & & & \text{teken onbeperk} \\ v_i &\geq 0 & i &= 1, 2, \dots, n \\ w_i &\geq 0 & i &= 1, 2, \dots, n. \end{aligned}$$