BA

stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE BESLISKUNDE BN 3/71 JUNI

BA

B.J. LAGEWEG

EEN BRANCH-AND-BOUND ALGORITHME VOOR MEERDIMENSIONALE KNAPSACKPROBLEMEN

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

Inhoud

1. Inleiding

Het geheeltallige lineaire programmeringsprobleem

(1) maximaliseer $\sum_{j=1}^{n} c_j x_j$

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

(3) $0 \leq x_j \leq h_j$ $\quad j = 1, \ldots, n$

(4) x_j geheel $\quad j = 1, \ldots, n$

(5) $c_j \geq 0, \quad b_i \geq 0, \quad a_{ij} \geq 0,$

ontmoet men onder meer bij de selectie van investeringsprojecten [18] en het snijprobleem [9], [17]. De oplossingsmethoden voor dit probleem vallen in twee groepen uiteen. De eerste groep lost het probleem op door dynamische programmering (Shapiro en Wagner [19]); bij een groot aantal bijvoorwaarden staat de dimensie van de toestandsruimte echter een succesvolle aanpak in de weg. De tweede groep lost het probleem op met branch-and-bound technieken [15], [16]. Hiertoe behoren de algorithme van Gilmore en Gomory [9] en de algorithme van Cabot [4]. Ook de hieronder beschreven algorithme valt in deze groep.
De kern van de branch-and-bound methode waarmee het meerdimensionale knapsack probleem hier aangepakt wordt, is afkomstig van Prof.dr. G. de Leve en Drs. J.Chr. van Dalen en omvat het splitsingsvoorschrift (§2) en de bepaling van de bovengrens (§3). Vervolgens (§4-8) worden mogelijkheden om de algorithme te versnellen onderzocht. De paragrafen 9 en 10 handelen over de programmering en de rekenresultaten van de algorithme. In §11 tenslotte wordt nagegaan of de algorithme ook bruikbaar is wanneer sommige coëfficiënten a_{ij} negatief zijn.

2. Het splitsingsvoorschrift

Het aantal toegelaten oplossingen van het probleem (1) - (5) is op grond van de formulering van het probleem eindig. Zij X een (deel)verzameling van toegelaten oplossingen. In X kan een variabele x_j slechts een eindig aantal waarden aannemen, zeg $L_j(X)$, $L_j(X)+1$, ..., $H_j(X)$. De verzameling van deze waarden noemen we het domein $D_j(X)$ van x_j in X:

$$D_j(X) \overset{\text{def}}{=} \{L_j(X),\ L_j(X)+1, \ldots,\ H_j(X)\},$$

$$\text{voor alle } j \in J \overset{\text{def}}{=} \{1,\ 2,\ \ldots,\ n\}.$$

Het aantal elementen van $D_j(X)$ geven we aan met $|D_j(X)|$. De verschilvariabele bij de i^{de} bijvoorwaarde $x_{n+i} \overset{\text{def}}{=} b_i - \sum_{j=1}^{n} a_{ij}\, x_j$ is ook begrensd in X:

$$L_{n+i}(X) \leq x_{n+i} \leq H_{n+i}(X), \qquad i = 1,\ \ldots,\ m.$$

Als we in het oorspronkelijke probleem (1) - (4) stellen

$$D_j(X) = \{0,\ 1,\ \ldots,\ h_j\},\ j \in J,$$

kunnen we dit probleem herformuleren als

$$\max Z = \sum_{j=1}^{n} c_j\, x_j$$

$$\text{onder} \quad \sum_{j=1}^{n} a_{ij}\, x_j \leq b_i \qquad i = 1,\ \ldots,\ m$$

$$(6) \qquad x_j \in D_j(X) \qquad j \in J$$

<u>Definitie 1.</u> Een variabele x_j is vrij in X als $|D_j(X)| > 1$, en gefixeerd in X als $|D_j(X)| = 1$.

Om een deelverzameling X te splitsen kiezen we een vrije variabele, zeg x_s, uit als splitsingsvariabele. We splitsen X in $|D_s(X)|$ deelverzamelingen $X_{L_s(X)}, \ldots, X_p, \ldots, X_{H_s(X)}$ door een deelverzameling X_p de volgende domeinen toe te kennen:

$$D_j(X_p) = \begin{cases} D_j(X), & j \in J, j \neq s \\ \{p\}, & j = s. \end{cases} \tag{7}$$

Door dit splitsingsvoorschrift behoudt het probleem in elke deelverzameling X_p de vorm (6) van het uitgangsprobleem. Bovendien zijn de deelverzameling X_p disjunct en vormt hun vereniging juist de gehele X.
Op de vraag welke vrije variabelen te kiezen als splitsingsvariabele, wordt ingegaan in §6.

3. De bepaling van de bovengrens

Het knelpunt bij de oplossing van probleem (1) - (4) is de eis "x_j geheel". Land en Doig [6] en Dakin [5] bepalen dan ook bovengrenzen door de voorwaarden (4) te laten vallen en het resterende continue probleem met lineaire programmering op te lossen. Men kan het probleem nog verder vereenvoudigen door de kriteriumfunctie te maximaliseren onder alleen de i^{de} bijvoorwaarde. Voor een verzameling X wordt dan het probleem:

$$B_i(X) = \max \sum_{j=1}^{n} c_j\, x_j$$

$$\text{onder} \sum_{j=1}^{n} a_{ij}\, x_j \leq b_i$$

(8) $$L_j(X) \leq x_j \leq H_j(X),\ j \in].$$

De optimale oplossing van (8) kan op een eenvoudige manier bepaald worden zonder gebruik van de simplexmethode. Ken daartoe allereerst aan elke variabele x_j de waarde $L_j(X)$ toe en rangschik de variabelen naar dalende c_j/a_{ij} , d.i. de bijdrage aan de kriteriumfunctie per eenheid "capaciteitsbeslag" ($c_j/a_{ij} = \infty$ als $a_{ij} = 0$). Hoog vervolgens de variabelen in deze volgorde op tot de bovengrens $H_j(X)$, totdat bij dit proces het linkerlid van de bijvoorwaarde het rechterlid overtreft; de variabele waarbij dit plaatsvindt, krijgt een zodanige waarde dat in de bijvoorwaarde het gelijkteken geldt. De nog resterende variabelen houden de waarde $L_j(X)$.
De scherpste aldus te berekenen bovengrens voor X is:

(9) $$B(X) \overset{\text{def}}{=} \min_{1 \leq i \leq m} B_i(X).$$

B(X) voldoet aan de gebruikelijke regels voor een bovengrens:

1) Als $\vec{x} = (x_1, \ldots, x_n) \in X$ dan $\sum_{j=1}^{n} c_j\, x_j \leq B(X)$;

2) Als $X_p \subset X$ dan $B(X_p) \leq B(X)$;

3) Als $X = \{\vec{x}\}$ dan $B(X) = \sum_{j=1}^{n} c_j\, x_j$.

Als $I(X) \subset \{1, \ldots, m\}$, dan voldoet $B'_{I(X)}(X) \stackrel{\text{def}}{=} \min_{i \in I(X)} B_i(X)$ wel aan 1) en 3), maar niet noodzakelijk aan 2). Dit euvel kan echter gemakkelijk worden verholpen: de bovengrens

$$B_{I(X_p)}(X_p) \stackrel{\text{def}}{=} \min\{B'_{I(X_p)}(X_p),\ B_{I(X)}(X)\}$$

voor een verzameling X_p die uit X is afgesplitst door één vrije variabele te fixeren, voldoet wel aan de 2e eis. Bij een geschikte keuze van $I(X)$ kan $B_{I(X)}(X)$ belangrijk sneller worden berekend dan $B(X)$ zonder veel aan scherpte te verliezen.

Omdat elke X eindig is, kan slechts een eindig aantal splitsingen worden uitgevoerd. De in §2 en 3 geschetste branch-and-bound algorithme levert derhalve na een eindige tijd de optimale oplossing van het probleem (1) - (4) op.

In de volgenden paragrafen worden verschillende manieren nagegaan om het branch-and-bound proces te versnellen. Het gemeenschappelijk element hierbij is het gebruik van informatie afkomstig van de optimale oplossing van het continue lineaire programmeringsprobleem (1) - (3), zoals ook gebeurt in algorithmes van Balas [2], Glover [10] en Geoffrion [8]. We noteren deze optimale oplossing als

$$(10) \qquad \begin{cases} z^c = x_0 + \sum_{j \in R} y_{0j}\, x_j & \\ & \\ y_{i0} = x_{B_i} + \sum_{j \in R} y_{ij}\, x_j, & i = 1, \ldots, m+n \end{cases}$$

met B, resp. R de indexverzameling van basis-, resp. niet-basis-variabelen in de optimale oplossing van (1) - (3).

4. Een geheeltallige beginoplossing

De bedoeling van de branch-and-bound aanpak is het aantal expliciet te onderzoeken oplossingen - d.w.z. deelverzamelingen $X = \{\vec{x}\}$ - zoveel mogelijk te beperken. Wanneer we een geheeltallige oplossing met kriteriumwaarde Z^* kennen, behoeft een deelverzameling X, waarvoor $B(X) < Z^*$, niet verder te worden onderzocht. Om snel over een "goede" geheeltallige oplossing, d.w.z. met een hoge kriteriumwaarde, te beschikken, kunnen we uitgaan van de naar beneden afgeronde oplossing van het continue probleem:

$$x_j = \begin{cases} [x_j]\ ^{*)}, & j \in B\cap] \\ 0, & j \in R\cap]. \end{cases}$$

Door de omgeving van de aldus geconstrueerde oplossing af te tasten, kan vaak een betere oplossing gevonden worden. Dit aftasten kan bijvoorbeeld systematisch gebeuren met een zoekprocedure van Hillier [14], die afwisselend nagaat:

- of nog vrije capaciteit beschikbaar is om een variabele met één te verhogen;
- of door gelijktijdige verandering van twee variabelen de kriteriumwaarde kan toenemen, terwijl de oplossing toegelaten blijft.

---

*) $[y] \stackrel{\text{def}}{=} \text{entier } (y)$.

5. Vermindering van het aantal relevante oplossingen in een verzameling oplossingen

Het potentiële aantal deelverzamelingen, waarin een verzameling van oplossingen van X gesplitst kan worden is $\prod_{j=1}^{n} |D_j(X)|$. Beperking van $|D_j(X)|$ verkleint het aantal mogelijke splitsingen en versnelt de oplossing van probleem (6) [23]. Een tweede voordeel is dat de bovengrens B(X) zeker niet zwakker maar wellicht sterker wordt als meer elementen uit een gegeven domein $D_j(X)$ verdwijnen. Voor de hoogste waarde die een variabele in X kan aannemen, geldt vanzelfsprekend

$$(11) \qquad H_j(X) \leq \min_{\substack{1 \leq i \leq m \\ a_{ij} > 0}} \{(b_i - \sum_{\substack{k=1 \\ k \neq j}}^{n} a_{ik} L_k(X))/a_{ij}\}, \; j = 1, \ldots, n.$$

Uit (10) volgt dat de waarde x_0 van een geheeltallige oplossing $\vec{x} = (x_1, \ldots, x_n)$ gelijk is aan

$$x_0 = Z^c - \sum_{j \in R} y_{0j} \, x_j .$$

Als we beschikken over een geheeltallige oplossing met waarde Z^*, moet voor een geheeltallige oplossing $\vec{x}$ met een hogere kriteriumwaarde - en alleen daarin stellen we belang - gelden:

$$(12) \qquad \sum_{j \in R} y_{0j} \, x_j < Z^c - Z^* .$$

Daaruit volgt:

$$(13) \qquad H_j(X) < (Z^c - Z^* - \sum_{\substack{k \in R \\ k \neq j}} y_{0k} \, L_k(X))/y_{0j} \qquad \text{voor } j \in R \text{ en } y_{0j} > 0.$$

Voor een basisvariabele x_{B_i} leiden we grenzen af uit

$$x_{B_i} = y_{i0} + \sum_{j \in R} (-y_{ij}) x_j .$$

We kunnen $H_{B_i}(X)$ overschatten door

(14) $$U_i(X) \overset{\text{def}}{=} \max y_{i0} + \sum_{j\in R} (-y_{ij})x_j$$

(15) $$\text{onder } \sum_{j\in R} y_{0j}\, x_j \leq Z^c - Z^*$$
$$L_j(X) \leq x_j \leq H_j(X),\ j \in R$$

en $L_{B_i}(X)$ onderschatten door

(16) $$V_i(X) \overset{\text{def}}{=} \max y_{i0} + \sum_{j\in R} y_{ij}\, x_j$$

onder (15).

$U_i(X)$ en $V_i(X)$ worden op dezelfde manier bepaald als $B_i(X)$, met dien verstande, dat variabelen x_j met een negatieve coëfficiënt in de kriteriumfunctie niet in aanmerking komen om opgehoogd te worden tot $H_j(X)$. Als $y_{ij} \neq 0$ in de i^{de} bijvoorwaarde, dan kan x_j geschreven worden als

$$x_j = \frac{y_{i0}}{y_{ij}} - \frac{1}{y_{ij}}\, x_{B_i} + \sum_{\substack{k\in R\\ k\neq j}} \left(-\frac{y_{ik}}{y_{ij}}\right) x_j.$$

We definieëren nu $U_{ij}(X)$ en $V_{ij}(X)$ resp. als

$$U_{ij}(X) \overset{\text{def}}{=} \max x_j,\ V_{ij}(X) = \max - x_j$$

$$\text{onder } \sum_{\substack{k\in R\\ k\neq j}} \{y_{0k} - \frac{y_{0j}y_{ik}}{y_{ij}}\}\, x_k - \frac{y_{0j}}{y_{ij}}\, x_{B_i} \leq Z^c - Z^* - \frac{y_{i0}y_{0j}}{y_{ij}}$$

$$L_k(X) \leq x_k \leq H_k(X),\quad k \in R,\ k \neq j$$

(17) $$L_{B_i}(X) \leq y_{B_i} \leq H_{B_i}(X).$$

Door variabelen met een negatieve coëfficiënt in de eerste bijvoorwaarde door hun tegengestelde te vervangen kunnen $U_{ij}(X)$ en $V_{ij}(X)$ op een analoge manier als $U_i(X)$ en $V_i(X)$ bepaald worden; dan geldt

(18) $$\max_{\substack{1 \leq i \leq m \\ y_{ij} \neq 0}} V_{ij}(X) = L_j(X) \leq x_j \leq H_j(X) \leq \min_{\substack{1 \leq i \leq m \\ y_{ij} \neq 0}} U_{ij}(X).$$

Uit (13), (15) en (17) volgt dat de scherpte van de beperkingen van de domeinen afhankelijk is van het verschil $Z^c - Z^*$: een tweede reden om snel een goede beginoplossing te construeren.

6. De keuze van de splitsingsvariabele

Uit $x_0 = Z^c - \sum_{j \in R} y_{0j} x_j$ volgt dat een oplossing een lagere waarde heeft, naarmate niet-basisvariabelen verder van nul, d.i. hun waarde in de optimale continue oplossing, afwijken. Voor "goede" geheeltallige oplossingen moeten we zoeken in de buurt van de optimale continue oplossing.

Voor de afwijking van de optimale continue oplossing voeren we de boete w_j (Driebeek [7]) in:

a) voor een niet-basisvariabele x_j: $w_j = y_{0j}$, d.i. de waarde van de corresponderende duale variabele;

b) als de i^{de} basisvariabele x_{B_i} een gehele waarde moet aannemen, neem dan voor w_{B_i} de minimaal noodzakelijke waardedaling van de kriteriumfunctie.

Als $x_{B_i} \leq [y_{i0}]$, ofwel $x_{B_i} + y_s = [y_{i0}]$, $x_s \geq 0$, dan

$$x_s - \sum_{j \in R} y_{ij} x_j = - f_{i0}, \quad \text{met } f_{i0} \overset{\text{def}}{=} y_{i0} - [y_{i0}].$$

Uit de duale simplextheorie en de eis "x_j geheel, $j \in J$" volgt de boete

$$P_i^L = \min_{j \in R} \begin{cases} y_{0j} f_{i0} / y_{ij} & y_{ij} > 0,\ j \notin J \\ \max\{y_{0j}, y_{0j} f_{i0}/y_{ij}\} & y_{ij} > 0,\ j \in J. \end{cases}$$

Als $x_{B_i} \geq [y_{i0}] + 1$, volgt

$$P_i^H = \min_{j \in R} \begin{cases} y_{0j}(1-f_{i0}) / (-y_{ij}), & y_{ij} < 0,\ j \notin J \\ \max\{y_{0j}, y_{0j}(1-f_{i0})/(-y_{ij})\} & y_{ij} < 0,\ j \in J. \end{cases}$$

De waardedaling is zeker gelijk aan $w_{B_i}^1 = \min\{P_i^L, P_i^H\}$.

Een andere mogelijkheid om de optredende waardedaling te berekenen (Tomlin [21]) volgt uit het feit dat uit de i^{de} voorwaarde, als x_{B_i} niet geheel is, de volgende Gomory-snede [12] afgeleid kan worden:

$$(19) \qquad \sum_{j \in R} f^*_{ij} \, x_j \geq f_{i0}$$

met

$$f^*_{ij} = \begin{cases} y_{ij} & y_{ij} \geq 0, \; j \notin] \\ - f_{i0} \, y_{ij} \; / \; (1-f_{i0}) & y_{ij} < 0, \; j \notin] \\ f_{ij} & f_{ij} \leq f_{i0}, \; j \in] \\ f_{i0}(1-f_{ij}) \; / \; (1-f_{i0}) & f_{ij} > f_{i0}, \; j \in] \end{cases}$$

Uit de duale simplextheorie volgt dat de waardedaling, bij onveranderde basis, gelijk is aan

$$P^G_i = \min_{j \in R} f_{i0} \, y_{0j} \; / \; f^*_{ij}.$$

Uit de formules voor P^L_i, P^H_i en P^G_i volgt: $P^G_i \geq \min(P^L_i, P^H_i)$.

We stellen: $w_{B_i} = P^G_i$.

Als splitsingsvariabele zoeken we een variabele, die een duidelijk onderscheid geeft tussen de bovengrenzen van de deelverzamelingen, zodat de niet direkt verder onderzochte deelverzamelingen een geringe kans hebben een optimale, of zelfs maar een "goede", oplossing te bevatten. We kiezen als splitsingsvariabele uit de vrije variabelen:

a) de niet-basisvariabele met de grootste boete, of anders

b) de basisvariabele met de grootste boete.

Een alternatieve regel is om onder b) te kiezen de basisvariabele met het grootste verschil tussen P^L_i en P^H_i.

Intuïtief aantrekkelijk is de regel: "kies in X als splitsingsvariabele een variabele x_s, waarvoor

$$|D_s(X)| = \min_{j \in]} \{ |D_j(X)| \; | \; |D_j(X)| > 1 \}",$$

zodat in iedere X een minimaal aantal deelverzamelingen voortgebracht wordt. Bij inperking van de domeinen volgens (13) stemt deze keuze van de splitsingsvariabele goeddeels overeen met de keuze volgens het boetekriterium.

7. Surrogaatvoorwaarden

De bepaling van de bovengrens B(X) vindt plaats door achtereenvolgens bovengrenzen $B_i(X)$ voor de kriteriumfunctie onder de i^{de} bijvoorwaarde te bepalen. De vraag is of de m bijvoorwaarden niet in één extra voorwaarde, een surrogaatvoorwaarde [11], samengevat kunnen worden. We beschouwen:

primaal probleem	duaal probleem
max $\vec{c}\,\vec{x}$	min $\vec{w}\vec{b} + \vec{y}\vec{h}$
onder $A\vec{x} \le \vec{b}$	onder $\vec{w}A + \vec{y} \ge \vec{c}$
$\vec{0} \le \vec{x} \le \vec{h}$ (20)	$\vec{w}\,,\ \vec{y} \ge \vec{0}$ (21)

Notatie. $\vec{v} = \vec{w}A + \vec{y} - \vec{c}$; $\vec{r} = \vec{h} - \vec{x}$;
B^1, resp R^1 is de verzameling basis-, resp. niet-basis-indices in de optimale oplossing van (20);
$\vec{x}^1$, resp. $(\vec{w}^2, \vec{y}^2, \vec{v}^2)$ is de optimale oplossing van (20), resp. (21).

Definitie 2. Een surrogaatvoorwaarde voor het primale probleem is een extra voorwaarde van de vorm $\vec{a}\vec{x} \le b_0$,
met $\vec{a} = \vec{w}A$, $b_0 = \vec{w}\vec{b}$, $\vec{w} \ge \vec{0}$.

We zoeken een surrogaatvoorwaarde die zoveel mogelijk dezelfde uitwerking heeft als de oorspronkelijke voorwaarden. Een maat voor deze uitwerking is:

Definitie 3. (Balas [2]): Een surrogaatvoorwaarde $\vec{a}^{*}\vec{x} \le b_0^{*}$ is sterker dan $\vec{a}\vec{x} \le b_0$, als geldt:

$$\max\{\vec{c}\vec{x} \mid \vec{a}^{*}\vec{x} \le b_0^{*},\ \vec{0} \le \vec{x} \le \vec{h}\} \le \max\{\vec{c}\vec{x} \mid \vec{a}\vec{x} \le b_0,\ \vec{0} \le \vec{x} \le \vec{h}\}.$$

De sterkste surrogaatvoorwaarde in de zin van Bals wordt gevonden door $\vec{w} = \vec{w}^2$ te nemen. Om dit te bewijzen vergelijken we:

$$\begin{array}{llcll} \max & \vec{c}\vec{x} & \qquad & \min & z\vec{w}^2\vec{b} + \vec{u}\vec{h} \\ \text{onder} & \vec{w}^2A\vec{x} \le \vec{w}^2\vec{b} & & \text{onder} & z\vec{w}^2A + \vec{u} \ge \vec{c} \\ & \vec{0} \le \vec{x} \le \vec{h} & & & z \ge 0 \\ \multicolumn{2}{l}{(\text{optimale oplossing } \vec{x}^3) \quad (22)} & & & \vec{u} \ge \vec{0} \quad (23) \end{array}$$

$z = 1$, $\vec{u} = \vec{y}^2$ is een toegelaten oplossing van (23), dus voor de optimale oplossing $(z^4, \vec{u}^4)$ van (23) geldt:

$$z^4\vec{w}^2\vec{b} + \vec{u}^4\vec{h} \le \vec{w}^2\vec{b} + \vec{y}^2\vec{h}.$$

Anderzijds is $\vec{w} = z^4\vec{w}^2$, $\vec{y} = \vec{u}^4$ een toegelaten oplossing van (21), waaruit volgt:

$$\vec{w}^2\vec{b} + \vec{y}^2\vec{h} \le z^4\vec{w}^2\vec{b} + \vec{u}^4\vec{h}.$$

Op grond van de dualiteitstheorie geldt dan voor $\vec{x}^1$ en $\vec{x}^3$:

$$\vec{c}\vec{x}^1 = \vec{c}\vec{x}^3.$$

Omdat $\max\{\vec{c}\vec{x} \mid \vec{a}\vec{x} \le b_0,\ \vec{0} \le \vec{x} \le \vec{h}\} \ge \vec{c}\vec{x}^1$, wordt voor $\vec{w} = \vec{w}^2$ dus de sterkste surrogaatvoorwaarde gevonden; het maximum van de kriteriumfunctie onder deze voorwaarde is zelfs niet groter dan onder de oorspronkelijke voorwaarden. Wegens $\vec{w}^2A \ge \vec{0}$ heeft deze voorwaarde dezelfde vorm als de oorspronkelijke voorwaarde en kan zij als $(m+1)^{ste}$ voorwaarde toegevoegd worden. Na herschrijving luidt zij:

$$\sum_{j=1}^{n} a_{m+1j}\, x_j \le \vec{c}\vec{x}^1 - \vec{y}^2\vec{h},$$

waarbij

$$(24) \qquad a_{m+1j} = \begin{cases} c_j & j \in B^1,\ x_j < h_j \\ c_j - y_j^2 & j \in B^1,\ x_j = h_j \\ c_j + v_j^2 & j \in R^1. \end{cases}$$

Deze voorwaarde geeft een scherpere bovengrens $B_{m+1}(X) < B(X)$, wanneer het aantal vrije variabelen groot is.

De hierboven afgeleide surrogaatvoorwaarde maakt geen gebruik van het feit dat voor een oplossing $\vec{x}$ van (1) - (4) moet gelden $\vec{c}\vec{x} \geq Z^*$, wanneer reeds een oplossing van (1) - (4) met kriteriumwaarde Z^* bekend is. De surrogaatvoorwaarde moet dan voldoen aan $\vec{a}\vec{x} - \vec{c}\vec{x} \leq b_0 - Z^*$. We kunnen als een andere maat voor de sterkte definiëren:

Definitie 4. (Geoffrion [8]): Een surrogaatvoorwaarde $(\vec{a}^* - \vec{c})\ \vec{x} \leq b_0^* - Z^*$ is sterker dan $(\vec{a} - \vec{c})\ \vec{x} \leq b_0 - Z^*$, wanneer:

$$\max\{b_0^* - (\vec{a}^* - \vec{c})\vec{x} \mid \vec{0} \leq \vec{x} \leq \vec{h}\}$$

$$\leq \max\{b_0 - (\vec{a} - \vec{c})\vec{x} \mid \vec{0} \leq \vec{x} \leq \vec{h}\}.$$

De sterkste surrogaatvoorwaarde volgens deze definitie wordt gevonden uit:

$$\min_{\vec{w} \geq \vec{0}} \Big[\max_{\vec{0} \leq \vec{x} \leq \vec{h}} \{\vec{w}\vec{b} - (\vec{w}A - \vec{c})\vec{x}\}\Big]$$

$$= \min_{\vec{w} \geq \vec{0}} \Big[\vec{w}\vec{b} + \max_{\vec{0} \leq \vec{x} \leq \vec{h}} \{(\vec{c} - \vec{w}A)\vec{x}\}\Big]$$

$$= \min_{\vec{w} \geq \vec{0}} \Big[\vec{w}\vec{b} + \min\{\vec{y}\vec{h} \mid \vec{y} \geq \vec{c} - \vec{w}A,\ \vec{y} \geq \vec{0}\}\Big]$$

$$= \min \Big[\vec{w}\vec{b} + \vec{y}\vec{h} \mid \vec{w}A + \vec{y} \geq \vec{c},\ \vec{w} \geq \vec{0},\ \vec{y} \geq \vec{0}\Big].$$

De optimale oplossing $(\vec{w}^2, \vec{y}^2)$ van het duale probleem levert ook de sterkste surrogaatvoorwaarde in de zin van Geoffrion op:

$$(\vec{w}^2A-\vec{c})\ \vec{x} \leq \vec{w}^2 b - Z^*$$

ofwel $$(\vec{w}^2A+\vec{y}^2-\vec{c})\ \vec{x} + \vec{y}^2\vec{r} \leq Z^c - Z^*$$

d.i.

(25) $$\sum_{j=1}^{n} v_j^2\ x_j + \sum_{j=1}^{n} y_j^2\ r_j \leq Z^c - Z^*.$$

De aldus verkregen surrogaatvoorwaarde is geschikt om de domeinen van de variabelen in te perken, waarbij de voorwaarde effectiever wordt naarmate een betere Z^* bekend is en meer variabelen gefixeerd zijn.

8. Toevoeging van sneden

De beperking van het aantal elementen van $D_j(X)$, $j \in R$ berust op het quotiënt $(Z^c - Z^* - \sum_{k \in R} y_{0k} L_k(X))/y_{0j}$.

Verkleining van dit quotiënt perkt het domein van niet-basisvariabelen in.

Aan de optimale oplossing van het continue probleem (1) - (3) kan de Gomory-snede (19) worden toegevoegd. Met de duale simplexmethode wordt de nieuwe optimale oplossing bepaald.

Door de toevoeging van een snede deelt de waarde van de kriteriumfunctie Z^c. Wanneer na één duale simplex-stap de optimale oplossing bereikt is, is de bovengrens voor een variabele x_j, $j \in R$, bepaald volgens (13), na transformatie kleiner dan voor transformatie, als

$$y_{0j} > \frac{f^*_{ij}}{f_{i0}} (Z^c - Z^* - \sum_{k \in R} y_{0k} L_k(X)).$$

Er zullen meer kleinere bovengrenzen gevonden worden, naarmate een betere Z^* beschikbaar is.

In de praktijk bleek dat de inperking van de domeinen door toevoeging van een of meer sneden òf niet optreedt, òf te verwaarlozen was, gezien het extra werk.

Toevoeging van sneden kan wel dienen om meerdere geheeltallige beginoplossingen te construeren, en wel door afronding van de successievelijk gevonden optimale continue oplossingen. Deze beginoplossingen vormen dan even zovele startpunten voor een heuristische zoekmethode naar een "goede" beginoplossing. Wanneer een heuristische methode afhankelijk is van het startpunt, zoals vaak het geval is [14], is de kans op een redelijk resultaat groter bij meerdere startpunten.

9. Beschrijving van het programma

Volgens de principes van de voorgaande paragrafen is een ALGOL-programma geschreven, dat verwerkt is op de EL-X8 van het Mathematisch Centrum. Enkele karakteristieken van dit programma zijn:

1) De gebruikte simplexalgorithme is een "straightforward algorithm" met bovengrenzen [13]. Zonder enig bezwaar zou echter ook een ander algorithme, zoals de productvorm algorithme gebruikt kunnen worden.

2) Het programma vertakt naar rechts, d.w.z. na splitsing van een verzameling X in deelverzamelingen $X_{L_s(X)}, \ldots, X_{H_s(X)}$ wordt eerst de juist gegenereerde deelverzameling X_{max} onderzocht, waarvoor

$$B(X_{max}) = \max_{L_s(X) \leq p \leq H_s(X)} B(X_p),$$

mits natuurlijk $B(X_{max}) > Z^*$. Het onderzoek van X gebeurt door een recursieve procedure.

3) De volgende inperkingen van de domeinen worden uitgevoerd:
   a) (11) voor elke deelverzameling X;
   b) (13), (14), (16) en (18), zodra een nieuwe geheeltallige oplossing Z_N^* wordt gevonden, die "belangrijk beter" is dan de oplossing Z_0^*, waarbij deze inperkingen voor het laatst zijn uitgevoerd. "Belangrijk beter" is gespecificeerd als $Z_N^* > Z_0^* + 0{,}4\ (Z^c - Z_0^*)$. De inperking wordt tegelijk uitgevoerd voor alle voortgebrachte deelverzamelingen, zeg X_0. Als $L_j(X_0) > H_j(X_0)$ voor een variabele x_j, dan is $X_0 = \emptyset$: Z_N^* is optimaal;
   c) (25) indirekt, door deze surrogaatvoorwaarde als 0^{de} voorwaarde aan de oorspronkelijke voorwaarden toe te voegen en voor elke X te verifiëren volgens (11), waarbij i in (11) dan loopt van 0 tot en met m.

4) Voor niet-basis-variabelen x_j is de ondergrens $L_j(X)$ meestentijds 0; voor basisvariabelen x_j op hun top, d.w.z. $x_j = h_j$ is de bovengrens $H_j(X)$ meestal h_j. In het programma worden deze grenzen niet

berekend, maar steeds op 0, resp. h_j gehouden.
De inperkingen volgens (17) worden vervangen door benaderingen.
Voor x_j, $j \in R$ geldt:

$$y_{ij}\, x_j = y_{i0} - x_{B_i} + \sum_{\substack{k\in R\\ k\neq j}} (-y_{ik})x_j,$$

waaruit volgt

$$V_i(X) - H_{B_i}(X) \leq y_{ij}\, x_j \leq U_i(X) - L_{B_i}(X).$$

Dan

$$(26) \qquad H_j(X) \leq \min \begin{cases} \min\limits_{\substack{1\leq i\leq m\\ y_{ij}>0}} (U_i(X) - L_{B_i}(X))/y_{ij} \\ \\ \min\limits_{\substack{1\leq i\leq m\\ y_{ij}<0}} (V_i(X) - H_{B_i}(X))/y_{ij}. \end{cases}$$

5) De omgeving van elke nieuwe geheeltallige oplossing wordt afgezocht door de heuristische zoekprocedure van §4.

6) In plaats van B(X), berekend volgens (9), gebruiken we als bovengrens $B_{I(X)}(X)$, waarbij I(X) op de volgende wijze wordt bepaald. Tot I(X) behoort in ieder geval (m+1), d.w.z. de surrogaatvoorwaarde (24). De overige voorwaarden i, waarvoor de ondergrens van de bijbehorende verschilvariabele $L_{n+i}(X) = 0$, worden als volgt gerangschikt:
- eerst voorwaarden i, waarvoor $(n+i) \in R$, naar dalende $y_{0,n+i}$;
- daarna voorwaarden i, waarvoor $(n+i) \in B$, naar stijgende $H_{n+i}(X)$.

In deze volgorde worden bovengrenzen $B_i(X)$ berekend, totdat tweemaal achtereen geen verbetering van $B_i(X)$ gevonden wordt. De idee achter deze bepaling van $B_{I(X)}(X)$ is de meest restrictieve voorwaarden eerst te bekijken en de niet-bindende voorwaarden over te slaan [20].

7) De invoer van de problemen moet de vorm hebben zoals beschreven in [1]. Het programma verwerkt zowel problemen van de vorm (1) - (4) als van de vorm

$$\min\{\vec{c}\vec{x} \mid A\vec{x} \geq \vec{b},\ \vec{0} \leq \vec{x} \leq \vec{h},\ A \geq 0,\ \vec{b} \geq \vec{0},\ \vec{c} \geq \vec{0}\}.$$

Bij de laatste soort problemen moet de bovengrens h_j bij iedere variabele zo zijn dat conversie tot een probleem van de vorm (1) - (4) doenlijk is.

10. Resultaten

Met het in de vorige paragraaf geschetste programma zijn 4 groepen problemen opgelost:

A) 7 testproblemen van Petersen [18] (A1-A7):
0-1 programmeringsproblemen. Uit n projecten met verwachte winst c_j moet een winstmaximaliserende keuze gedaan worden, waarbij in periode i een bedrag b_i beschikbaar is en project j een investering a_{ij} vergt;

B) 9 testproblemen van Trauth en Woolsey [22] (B1-B9): toewijzingsproblemen van 10 variabelen en 1 bijvoorwaarde;

C) 5 testproblemen van Haldi [22] (C1-C5): minimaliseringsproblemen onder $\geq$-bijvoorwaarden, waarbij de matrix a_{ij} alleen nullen en enen bevat;

D) 18 gegenereerde problemen (D1-D18) van de vorm (1) - (4), waarin gemiddeld 15% van de a_{ij} nul is en b_i gemiddeld 1,2 $\sum_{j=1}^{n} a_{ij}$.

Deze problemen zijn eveneens opgelost met een programma volgens de algorithme van Land en Doig [6].

Tabel 1. Vergelijking rekentijden

probleem	n×m	Z*	Multiknapsack rekentijd [1]	Multiknapsack Z^i [2]	Land and Doig rekentijd [1]	Land and Doig Z^i [2]
A1	6×10	3800	1,5		9,5	
A2	10×10	8706,1	3,9		40,0	
A3	15×10	4015	9,3		> 180	4005
A4	20×10	6120	8,4		> 180	6120
A5	28×10	12400	15,7		> 180	12080
A6	39× 5	10618	67,1		> 180	10338
A7	50× 5	16537	236,8		> 180	14109
B1	10× 1	50	0,7		3,0	
B2	10× 1	52	1,0		7,8	
B3	10× 1	57	1,3		6,0	
B4	10× 1	62	1,6		7,4	
B5	10× 1	67	0,5		7,2	
B6	10× 1	68	1,0		3,7	
B7	10× 1	70	1,4		7,7	
B8	10× 1	75	3,1		4,5	
B9	10× 1	85	0,5		5,9	
C1	7× 7	27	1,0		3,3	
C2	7× 7	28	1,1		17,5	
C3	15×15	80	3,9		48,3	
C4	15×15	105	>180	105	> 180	105
C5	15×35	6	39,9		> 180	6
D1	12×10		2,6		18,8	
D2	12×10		2,8		51,4	
D3	12×14		2,7		18,0	
D4	12×14		3,7		84,0	
D5	12×18		2,3		20,3	
D6	12×18		5,8		177,9	
D7	20× 2	185,1	1,3		2,4	
D8	20× 2	371	1,7		14,8	
D9	20×10	1064	31,4		> 180	1054,4
D10	20×10	504,7	3,7		65,1	
D11	20×30	182,1	7,5		116,7	
D12	20×30	227,4	23,1		> 180	223,4
D13	50× 2	16156,2	8,2		127,2	16156,2
D14	50× 2	11758,1	12,2		> 180	11745,5
D15	50×25	2607,2	25,9		> 180	2607,2
D16	50×25	1432,9	163,6		> 180	1422,7
D17	50×60	494,3	36,0		>1000	494,3
D18	50×60	337,8	75,0		>1000	- [3]

[1] in seconden

[2] waarde beste geheeltallige oplossing bij overschrijding toegestane rekentijd

[3] nog geen oplossing gevonden.

Uit de tabellen blijkt dat de multiknapsack algorithme (MKA) voor alle vier groepen problemen sneller werkt dan de algorithme van Land en Doig (ALD). Voor problemen die beide algorithmes binnen 3 minuten oplossen is de MKA een factor 20 à 25 sneller. Deze factor neemt toe met de omvang van het probleem; bij de één-dimensionale knapsackprobleem (serie B) van geringe omvang is het verschil in rekentijd klein, bij de 50×60 problemen van serie D is de ALD na een kwartier nog niet beëindigd, terwijl de MKA gemiddeld binnen de minuut een optimale oplossing gevonden en geverifieerd heeft.

Zowel bij de MKA als bij de ALD treden grote fluctuaties in rekentijden op voor problemen van dezelfde omvang, er zijn moeilijke en makkelijke problemen. Gebruikt men bij de MKA een ander kriterium voor de keuze van de splitsingsvariabele, bv. een willekeurige volgorde, dan vertonen de fluctuaties de neiging groter te worden. Een verklaring hiervoor kan zijn dat men in dat geval de "richting" van het branch-and-bound proces minder in de hand heeft. Daardoor loopt men kans in een gebied te gaan zoeken waar alleen oplossingen voorkomen, die weliswaar een verbetering betekenen, maar die nog ver van de optimale oplossing afliggen. Bij de rekenexperimenten bleek de keuze van de splitsingsvariabelen de factor met de grootste invloed op de rekentijden.

Het MKA-programma heeft eveneens een aantal zuivere knapsackproblemen opgelost, d.w.z. met 1 bijvoorwaarde en uitsluitend 0-1 variabelen (tabel 2).

Tabel 2. Eendimensionale 0-1 knapsackproblemen

n	rekentijd MKA [1]		
	$b_1 = \frac{1}{3} \sum a_{1j}$	$b_1 = \frac{1}{2} \sum a_{1j}$	$b_1 = \frac{2}{3} \sum a_{1j}$
10	0,8	0,8	0,6
20	1,2	1,3	1,3
30	2,2	4,6	4,4
40	4,9	3,3	3,5
50	6,5	5,6	4,3
100	17,1	13,8	18,9
150	49,9	37,3	27,5
200	48,7	71,2	71,1
250	84,5	70,9	76,7
300	173,3	99,3	103,4
500	482,8	273,1	290,0

1) in seconden.

We merken op:

- uit de tabel kan niet afgeleid worden dat de rekentijd bij dit soort problemen beïnvloed wordt door de grootte van het rechterlid;
- op het onderzochte interval geldt het volgende verband tussen rekentijd en aantal variabelen: $t = 0{,}011\ n^{1,6}$.

11. Negatieve coëfficiënten a_{ij}

Wanneer getallen $a_{ij} < 0$ in de bijvoorwaarden voorkomen, kan de opzet van de boven beschreven algorithme worden aangehouden; op een aantal onderdelen moeten echter wijzigingen worden aangebracht:

1) Bij de bepaling van $B_i(X)$ door het probleem (8) op te lossen bestaat de eerste stap uit het toekennen van de waarde $H_j(X)$ aan variabelen x_j met $a_{ij} < 0$. Verder kan dezelfde werkwijze worden gevolgd.
2) De naar beneden afgeronde continue oplossing (§4) kan niet toegelaten zijn. Gewoonlijk is echter in de omgeving van deze oplossing op een eenvoudige manier een toegelaten oplossing te vinden.
3) De inperking van de domeinen volgens (11) wordt, als L_j en H_j de scherpste tot nu toe bekende grenzen zijn:

$$H_j(X) \leq \min_{\substack{1\leq i\leq m \\ a_{ij}>0}} \{(b_i + \sum_{\substack{1\leq k\leq n \\ a_{ik}<0}} (-a_{ik})H_k - \sum_{\substack{1\leq k\leq n \\ a_{ik}>0 \\ k\neq j}} a_{ik} L_k)/a_{ij}\} \tag{27}$$

$$L_j(X) \geq \max_{\substack{1\leq i\leq m \\ a_{ij}<0}} \{(b_i + \sum_{\substack{1\leq k\leq n \\ a_{ik}<0 \\ k\neq j}} (-a_{ik})H_K - \sum_{\substack{1\leq k\leq n \\ a_{ik}>0}} a_{ik} L_k)/a_{ij}\}$$

   Wanneer men nu na inperking variabelen gaat fixeren is men er niet langer zeker van dat een willekeurige aanvulling van de partiële oplossing een toegelaten oplossing oplevert. Het is zaak de keuze van de splitsingsvariabele tijdig te sturen in de richting van een toegelaten aanvulling.
4) In de surrogaatvoorwaarde (24) $\vec{w}^2 A\vec{x} \leq \vec{w}^2\vec{b}$ geldt niet langer $\vec{w}^2 A \geq \vec{0}$; ook in deze voorwaarde kunnen coëfficiënten $a_{m+1,j} < 0$ optreden, en wel voor variabelen die in de continue oplossing op hun bovengrens zitten.

Met een programma, geschikt voor $a_{ij} < 0$ zijn 30 gegenereerde problemen opgelost, in omvang variërend tussen $n \times m = 6 \times 2$ tot 12×18, met gemiddeld 13% van de $a_{ij} < 0$ (categorie I). Tevens zijn 30 problemen van dezelfde omvang opgelost met uitsluitend $a_{ij} \geq 0$ (categorie II).

Tabel 3. Resultaten voor $a_{ij} < 0$

Categorie problemen	gemiddelde rekentijd [1]		verhouding rekentijd ALD : MKA
	MKA	ALD	
I : sommige $a_{ij} < 0$	5,1	38,6	7,6
II : alle $a_{ij} \geq 0$	1,8	25,0	13,8
verhouding rekentijd I:II	2,8	1,5	

[1] in seconden.

Uit de tabel blijkt (kolom 3) dat de MKA ook voor problemen van categorie I sneller werkt dan de ALD, maar dat de relatieve tijdwinst nog maar de helft is van de tijdwinst bij problemen van categorie II.
Voorts zien we (regel 3) dat de toename in rekentijd voor het MKA ongeveer 2× zo sterk is als voor het ALD, wanneer men problemen met resp. zonder $a_{ij} < 0$ oplost.
Voor iets grotere problemen (20 à 30 variabelen) is deze toename nog veel sterker, voor 20 variabelen liggen de rekentijden al rond de 2 minuten (vergelijk tabel 1).
Een verklaring voor de relatief ongunstige prestaties bij problemen van categorie I moet gezocht worden in de omstandigheden dat:

- de bovengrenzen $B_i(X)$ minder scherp worden naarmate meer $a_{ij} < 0$ optreden (zie punt 11.1);
- de inperking van de domeinen volgens (27) minder effectief is, wegens de term $\sum_{a_{ik}<0} (-a_{ik})H_k$ in het rechterlid;
- de mogelijkheid bestaat dat deelverzamelingen zonder toegelaten oplossingen verder onderzocht worden (punt 11.3).

Summary

In this paper the multidomensional knapsack problem is solved by a branch-and-bound method.

Using the optimal solution of the continuous L.P.-problem, one can:

- obtain an integer solution at the beginning of the algorithm;
- diminish the number of relevant solutions in a subset of solutions;
- calculate penalties for discerning "good" and "bad" subsets;
- construct surrogate constraints replacing the original ones.

The algorithm compares favourably with the algorithm of Land and Doig.

Literatuur

[1] Anthonisse, J.M., Een invoerbeschrijving voor continue lineaire programmering (gefotokopieerde aantekening).

[2] Balas, E., Discrete programming by the filter method, Opns. Res. 15, 915-957 (1967).

[3] Balinski, M., Integer programming: Methods, uses, computation, Man. Sc. 12, 253-313 (1965).

[4] Cabot, A., An enumeration algorithm for knapsack problems, Opns. Res. 18, 306-311 (1970).

[5] Dakin, R.J., A tree search algorithm for mixed integer programming problems, The Computer Journal 8, 250-255 (1965).

[6] Doig, A.G. en A.H. Land, An automatic method of solving discrete programming problems, Econometrica 28, 497-520 (1960).

[7] Driebeek, N., An algorithm for the solution of mixed integer programming problems, Man. Sc. 12, 576-587 (1966).

[8] Geoffrion, A.M., An improved inplicit enumeration approach for integer programming, Opns. Res. 17, 437-454 (1969).

[9] Gilmore, P.C. en R.E. Gomory, A linear programming approach to the cutting-stock problem, Part II, Opns. Res. 11, 863-888 (1963).

[10] Glover, F., A multiphase-dual algorithm for the zero-one programming problem, Opns. Res. 13, 879-919 (1965).

[11] Glover, F., Surrogate constraints, Opns. Res. 16, 741-749 (1968).

[12] Gomory, R.E., An algorithm for the mixed integer problem,
The Rand Corporation RM-2597, Santa Monica, California (1960).

[13] Hadley, G., Linear Programming,
Addison-Wesley, Reading, Massachusetts (1962).

[14] Hillier, F., Efficient heuristic procedures for integer linear programming with an interior,
Opns. Res. 17, 600-637 (1969).

[15] Lawler, E.L. en D.E. Wood, Branch-and-bound methods: a survey,
Opns. Res. 14, 699-719 (1966).

[16] Mitten, L.G., Branch-and-bound methods: general formulation and properties,
Opns. Res. 18, 24-35 (1970).

[17] Noord, P.F. de, Een snijprobleem (scriptie),
Amsterdam (1970).

[18] Petersen, C., Computational experience with variants of the Balas algorithm applied to the selection of R & D projects,
Man. Sc. 13, 736-750 (1967).

[19] Shapiro, J.F. en H.M. Wagner, A finite renewal algorithm for the knapsack and turnpike models,
Opns. Res. 15, 319-341 (1967).

[20] Thompson, G.L., F.M. Tonge en S. Zionts, Techniques for removing nonbinding constraints and extraneous variables from linear programming problems,
Man. Sc. 12, 588-608 (1966).

[21] Tomlin, J., Branch-and-bound methods for integer and non-convex programming, in:
J. Abadie (ed.), Integer and non-linear programming,
North-Holland Publishing Company, Amsterdam (1970).

[22] Trauth, C.A. en R.E. Woolsey, Integer linear programming: a study in computational efficiency,
Man. Sc. 15, 481-493 (1969).

[23] Zionts, S., Towards an unifying theory for integer linear programming,
Opns. Res. 17, 359-367 (1969).