Z. Wahrscheinlichkeitstheorie verw. Geb. 23, 255–259 (1972)

Ganzzahlige lineare Programmierung und das Knapsack-Problem

Beat Schmid

Es existieren gute Algorithmen für spezielle ganzzahlige lineare Programme, etwa für Netzwerkflußprobleme, oder – wie es scheint – für das Knapsack-Problem. Für das allgemeine ganzzahlige lineare Programm allerdings ist die Situation weniger günstig. Es ist deshalb wohl nützlich, spezielle Typen von ganzzahligen Linearprogrammen zu suchen, in die alle übrigen Linearprogramme transformiert werden können: Man kann dann nach jenen Algorithmen suchen, die, da sie die spezielle Struktur des Problems ausnutzen können, möglicherweise besser, jedenfalls einfacher sind. Hier soll gezeigt werden, daß jedem ganzzahligen linearen Programm ein Knapsack-Problem zugeordnet werden kann. Es wird gezeigt, wie eine große Klasse von Knapsack-Algorithmen als Algorithmen für Boolesche Linearprogramme aufgefaßt werden kann.

I

Die Aufgabe

$$(\mathrm{LP}) \quad \begin{cases} c' x \Rightarrow \min \\ A x = a \\ x \geqq 0, \quad x \in R^n, \end{cases}$$

wo $c \in R^n$, $a \in R^m$ und A eine (m, n)-Matrix mit reellen Koeffizienten ist, heißt *lineares Programm* (LP). R ist die Menge der reellen Zahlen.

Ganzzahliges lineares Programm (GLP) nennen wir hier die Aufgabe

$$(\mathrm{GLP}) \quad \begin{cases} (1) & c' x \Rightarrow \min \\ (2) & A x = a \\ (3) & 0 \leqq x \leqq s, \end{cases}$$

wo c, s, $x \in Z^n$, $a \in Z^m$, A eine (m, n)-Matrix mit Koeffizienten aus Z ist. Z ist die Menge der ganzen Zahlen. Ein GLP mit $s' = e' = (1, 1, \ldots, 1)$ heißt *Boolesches Linearprogramm* (BLP). Ein sehr einfaches BLP ist das *Knapsack-Problem* (KP):

$$(\mathrm{KP}) \quad \begin{cases} c' x \Rightarrow \min \\ a' x \leqq b, \quad x \in \{0, 1\}^n, \ a, c \in Z^n. \end{cases}$$

Ein Vektor $x \in Z^n$ heißt *zulässige Lösung* eines GLP, wenn x (2) und (3) erfüllt. Eine zulässige Lösung heißt *optimal*, wenn sie (1) minimiert.

Zwei Programme

$$\text{P1:}\quad \min\{f(x)|x\in S\}$$
$$\text{P2:}\quad \min\{g(x)|x\in T\}$$

heißen *äquivalent*, wenn eine Bijektion $\varphi\colon S\to T$ existiert mit $f(x)=g(\varphi(x))$ für alle $x\in S$.

Bemerkung. Jedem GLP läßt sich ein äquivalentes BLP zuordnen: Man setze

$$x_i=\sum_{j=0}^{n_i}2^j x_{ij},\quad \text{wo}\quad n_i=[\mathrm{ld}\, s_i]+1$$

ist. (ld y ist der Logarithmus zur Basis 2 von y, $[a]$ ist die größte ganze Zahl in der reellen Zahl a.) Damit geht das GLP in n Variablen in ein BLP in $N=\sum_{i=1}^{n} n_i$ Variablen über, $x_{ij}\in\{0,1\}$.

II

Es soll nun jedem GLP ein äquivalentes KP zugeordnet werden. Dies scheint die Suche nach Algorithmen etwas zu vereinfachen; denn jeder GLP-Algorithmus muß mindestens auch ein Algorithmus für das Knapsack-Problem sein, aber nicht umgekehrt.

Lemma. *Das System*

$$\text{S1}\quad \begin{cases} a'x=c \\ b'x=d \\ x\in P\subset Z^n \end{cases}$$

ist äquivalent mit dem System

$$\text{S2}\quad \begin{cases} (a+\tau b)'x=c+\tau d \\ x\in P. \end{cases}$$

Dabei ist $a, b\in Z^n$; $c, d\in Z$, P eine beliebige, aber beschränkte Teilmenge von Z^n und $\tau>\max\{|a'x|\,|x\in P\}+|c|$.

Beweis. (1) x sei Lösung von S1 $\Rightarrow$ x Lösung von S2.

(2) Sei $\hat{x}$ Lösung von S2.

$\Rightarrow (a'\hat{x}-c)+\tau(b'\hat{x}-d)=0$. Sei $b'\hat{x}-d=k\neq 0$,

$\Rightarrow k$ ist ganz und deshalb ist $|a'\hat{x}-c|\geqq\tau \Rightarrow |a'\hat{x}|\geqq\tau-|c|$,

$\Rightarrow |a'\hat{x}|>\max\{|a'x|\,|x\in P\}$, was nicht möglich ist, da $\hat{x}\in P$ ist. Also muß $b'\hat{x}-d=0$ sein, d.h. $\hat{x}$ ist Lösung von S1.

Korollar. *Jedem linearen System*

$$Ax=a,$$
$$0\leqq x\leqq s$$

$x, s\in Z^n$, $a\in Z^m$, $A(m,n)$-Matrix, läßt sich ein äquivalentes System

$$b'x=d,$$
$$0\leqq x\leqq s$$

$b, s, x\in Z^n$, $d\in Z$, zuordnen.

Beweis. Folgt mit $m-1$-maliger Anwendung des Lemmas.

Satz 1. *Jedem* GLP

$$c' x \Rightarrow \min,$$
$$A x = d,$$
$$0 \leqq x \leqq s$$

$c, s, x \in Z^n$, $d \in Z^m$, A (m, n)-*Matrix, läßt sich ein äquivalentes* KP *zuordnen:*

$$(\bar{c} - \rho\, \bar{a})' x \Rightarrow \min,$$
$$\bar{a}' x \leqq b, \quad x \in \{0, 1\}^N,$$

wo $\bar{c}, \bar{a} \in Z^N$, $b \in Z$, $N = \sum_{i=1}^{n} n_i$, $n_i = [\mathrm{ld}\, s_i] + 1$, *ist,* $\rho > \sum_{i=1}^{n} |\bar{c}_i|$ (s_i *und* c_i *sind Komponenten von* s *bzw.* c_i.)

Beweis. Mit obigem Korollar erhält man zunächst ein zum GLP äquivalentes Programm

$$c' x \Rightarrow \min,$$
$$a' x = b,$$
$$0 \leqq x \leqq s$$

$a \in Z^n$, $b \in Z$. Ersetzt man hierin die x_i durch die in der Bemerkung am Schluß von I angegebenen Ausdrücke, so erhält man ein BLP

$$\bar{c}' x \Rightarrow \min,$$
$$\bar{a}' x = b,$$
$$x \in \{0, 1\}^N, \quad N = \sum_{i=1}^{n} n_i.$$

Bleibt noch zu zeigen, daß dieses BLP mit dem KP von Satz 1 äquivalent ist. Sei $\hat{x}$ optimale Lösung des BLP. Dann ist $\hat{x}$ zulässig in KP. Sei $\bar{x}$ optimale Lösung des KP. Es gilt also: $(\bar{c} - \rho\, \bar{a})' \bar{x} = (\bar{c} - \rho\, \bar{a})' \hat{x} \Leftrightarrow (\bar{c} - \rho\, \bar{a})' x + \rho\, b = (\bar{c} - \rho\, \bar{a})' \hat{x} + \rho\, b =$ $\bar{c}' \hat{x} + \rho(b - \bar{a}' \hat{x}) = \bar{c}' \hat{x}$, d.h. $\bar{c}' \bar{x} + \rho\, k = \bar{c}' \hat{x} \Leftrightarrow \rho\, k = \bar{c}' (\hat{x} - \bar{x}) \leqq \sum_{i=1}^{N} |c_i| < \rho$, d.h. $\rho\, k < \rho \Leftrightarrow (k-1)\, \rho < 0$. $k = b - \bar{a}' \hat{x}$. ρ ist positiv, k nicht negativ und ganz, folglich muß $k = 0$ sein, d.h. aber: $\bar{x}$ ist optimale Lösung des BLP (falls dieses überhaupt eine Lösung besitzt).

III

Da jedem GLP ein äquivalentes KP zugeordnet werden kann, kann jeder Algorithmus für dieses für die ganzzahlige Programmierung benutzt werden: das zum GLP äquivalente KP kann explizite berechnet werden. Für eine wichtige Klasse von Algorithmen läßt sich diese Berechnung jedoch umgehen.

Sei R ein geordneter Ring. Ein *R-Algorithmus* $A(R, P)$ sei ein Algorithmus, der in R rechnet, d.h. neben den logischen Operationen sind Addition, Multiplikation und Vergleich von Ringelementen definiert. $P \subset R$ sei eine Parametermenge.

Ein R-Algorithmus heißt *beschränkt*, falls die Anzahl der Operationen Addition, Multiplikation und Vergleich, die nötig sind, um die optimale Lösung zu finden, für jede Wahl von P beschränkt ist durch eine Zahl K.

Die Menge der beschränkten Z-Algorithmen (Z = Ring der ganzen Zahlen) ist recht umfangreich: jeder Q-Algorithmus (Q = Ring der rationalen Zahlen) kann als Z-Algorithmus aufgefaßt werden. Jeder Algorithmus kann durch Hinzufügen einer Stoppanweisung zu einem beschränkten gemacht werden (Abbruch z. B. nach m Operationen).

Satz 2. *Sei* $A(Z; a_i, c_i, b\ (i=1, 2, \ldots, n))$, *kurz* $A(Z)$, *ein beschränkter Z-Algorithmus für das* KP

$$c' x \Rightarrow \min,$$
$$a' x = b, \quad x \in \{0, 1\}^n$$

(a_i, c_i *sind Komponenten von* a, c). *Dann ist* $A(Z[t]; \alpha_i, \gamma_i, \beta\ (i=1, 2, \ldots, n))$, *kurz* $A(Z[t])$, *ein beschränkter* $Z[t]$*-Algorithmus für das* BLP

$$d' x \Rightarrow \min,$$
$$A x = f, \quad x \in \{0, 1\}^n,$$

$d \in Z^n$, $f \in Z^m$, A (m, n)*-Matrix mit ganzen Koeffizienten.* $Z[t]$ *ist der Polynomring über* Z *mit der „lexikographischen" Ordnung:*

$$\pi = \sum_{i=1}^{k} p_i t^{i-1} \leqq \sigma = \sum_{i=1}^{h} s_i t^{i-1} \Leftrightarrow p_L \leqq s_L,$$

wo $L = \max\{i | p_i \neq s_i\}$ *ist. Weiter ist*

$$\alpha_i = \sum_{j=1}^{m} a_{ji} t^{j-1}, \quad \gamma_i = d_i - \rho\, \alpha_i, \quad \rho > \sum_{j=1}^{n} |d_j|, \quad \beta = \sum_{j=1}^{m} f_j t^{j-1}$$

(d_i, f_j *sind Komponenten von* d, f; a_{ij} *sind Koeffizienten von* A).

Beweis. (1) In Satz 1 wird jedem BLP ein äquivalentes KP zugeordnet. Dabei wird eine Restriktion sukzessive mit Zahlen

$$\tau_i > \max\left\{\left|\sum_{j=1}^{n} a_{ij} x_j\right| \middle| x \in \{0, 1\}^n\right\} + |f_i|$$

multipliziert und dann die i-te Restriktion dazuaddiert, für $i = m-1, m-2, \ldots, 1$. Wählt man $\tau_i = t$ für $i = 1, 2, \ldots, m-1$, t groß genug, so sind die Koeffizienten des entstehenden KP „Polynome" in t.

(2) Da der Algorithmus $A(Z)$ beschränkt ist, bleiben die Koeffizienten der „Polynome", die in $A(Z)$ entstehen, beschränkt. K sei eine solche Schranke.

(3) Sei $M_Z \subset Z$ die (endliche) Menge der Zahlen, die $A(Z)$ aus a_i, b, c_i produziert, entsprechend sei $M_{Z[t]} \subset Z[t]$ die Menge der Polynome, die $A(Z[t])$ aus $\alpha_i, \beta, \gamma_i$ erzeugt. Die Parameter von $A(Z)$ seien dargestellt als „Polynome" in der Zahl t. Dann können alle Zahlen in M_Z als „Polynome" in t dargestellt werden. Falls die Parameter von $A(Z)$ und $A(Z[t])$ je gleiche Polynome sind, sind M_Z und $M_{Z[t]}$ isomorph bezüglich Addition und Multiplikation. Diese Isomorphie

wird gestiftet durch die Abbildung

$$\tau_t\colon\ p=\sum_{i=1}^{k} p_i t^{i-1}\in M_Z \to \pi=\sum_{i=1}^{k} p_i t^{i-1}\in M_{Z[t]}.$$

Falls $t\geqq 2K$ ist, ist τ_t auch ordnungstreu, d.h.

$$p\leqq q \Leftrightarrow \tau_t(p)=\tau_t(q) \Leftrightarrow \sum_{i=1}^{k} p_i t^{i-1}\leqq \sum_{i=1}^{k} q_i t^{i-1} \Leftrightarrow \sum_{i=1}^{L}(q_i-p_i)\,t^{i-1}\geqq 0,$$

$$L=\max\{i\mid p_i\neq q_i\},\qquad \Leftrightarrow (q_L-p_L)\,t^{L-1}+\sum_{i=1}^{L-1}(q_i-p_i)\,t^{i-1}\geqq 0.$$

Falls $\sum_{i=1}^{L-1}(q_i-p_i)\,t^{i-1}<t^{L-1}$ ist, muß dann, wie gefordert, $p_L<q_L$ sein. Die letzte Ungleichung gilt jedoch immer, wenn $t\geqq 2K>3$ ist: Sei $\sum_{i=1}^{L} s_i t^{i-1}$ ein beliebiges Polynom mit $|s_i|\leqq K$ und $|s_L|\neq 0$. Dann ist in der Tat

$$\left|\sum_{i=1}^{L-1} s_i t^{i-1}\right|\leqq \sum_{i=1}^{L-1}|s_i|\,t^{i-1}\leqq \sum_{i=1}^{L-1}\frac{t}{2}\cdot t^{i-1}=\frac{1}{2}\sum_{i=1}^{L-1} t^i=\frac{1}{2}\left(\frac{t^L-1}{t-1}-1\right)<t^{L-1}.$$

Die letzte Ungleichung ist gleichbedeutend mit

$$t^L-t<2t^L-2t^{L-1} \Leftrightarrow 2t^{L-1}-t<t^L$$

$$\Leftarrow t^L=t\cdot t^{L-1}>(2+1)\,t^{L-1}=2t^{L-1}+t^{L-1}\geqq 2t^{L-1}\qquad \text{für } t>3.$$

Wenn man also in $A(Z[t])$ rechnet, so kann man mit τ_t^{-1} nach $A(Z)$ gehen: in $A(Z)$ löst man dann gerade das gemäß Satz 1 zum BLP äquivalente KP. Wenn $A(Z)$ dieses KP löst, dann ist $A(Z[t])$ ein Algorithmus für das BLP – wie behauptet.

IV

Eine zu Satz 1 analoge Aussage läßt sich für eine Klasse von Optimierungsaufgaben machen, die alle ganzzahligen Programme umfaßt. Für eine große Menge von nichtlinearen Optimierungsaufgaben läßt sich auch eine Satz 2 entsprechende Aussage beweisen. Diese Resultate werden später veröffentlicht.

Wie der Autor nachträglich am 7th Mathematical Progr. Symposion in Den Haag von Herrn Elmaghraby erfahren hat, ist das in dieser Arbeit angegebene Lemma für Restriktionen mit nichtnegativen Koeffizienten schon im letzten Jahrhundert von Mathew bewiesen worden.

Beat Schmid
Institut für Operations Research
Weinbergstr. 59
CH-8006 Zürich
Schweiz

(Eingegangen am 9. Juli 1971)