経営科学 第12巻 第4号 (1969年8月)

0—1 整数計画問題の一解法†

阿部健一*
木村正行*

1. まえがき

すべての変数が0または1の値のみをとるような L.P 問題(0—1整数計画問題という)の解法については,いろいろな立場からの数多くの研究がなされており,とくに著名なものとして,R. Gomory の cutting-plane method[6], E. Balas の Additive Algorithm[1] があげられる.前者は解析的手法とでもよぶべき解法で,シンプレックス法を基本とする.一方,後者の Additive Algorithm はバックトラックの原理[3] を用いて,一つ一つの解を直接的あるいは間接的に調べるという手法で,アルゴリズムの構成は非常に簡単である.また,解法で必要となる算法が加減算のみであるため, Additive という名称が与えられた.両者の解法としての効力を比較してみると,与えられた問題の性質(主に係数マトリックスの特徴など)に応じ,しばしば異なった結果をうる.したがって,両者の優劣は一概には判断できない.しかし, Additive Algorithm にはアルゴリズムの効力を高める手段が多数残されており,今後の研究に待つところが大きいように思われる.

本論文では, Additive Algorithm と同様のバックトラックの原理による解法を与える.解法の構成をより一般的にするため,制限関数という概念を導入する.

まず2節では,与えられた0—1整数計画問題を正係数の制約式からなる問題に変換することを述べる.3節で解法のベースとなる解系列の定義を与え,4節で制限関数の定義および解系列を用いた解法を与える.5節では,制限関数の具体的な三つの構成例を示す.6節では,二つの簡単な例題を示す.

2. 0—1整数計画問題

0—1整数計画問題は一般につぎのように与えられる.

問題 IP. 1 (2), (3)の制約のもとに(1)を最小にせよ.

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

† 1969年1月4日受理
* 東北大学電気通信研究所

(2) $\sum_{j=1}^{n} a_{ij}\ x_j \leqq b_i$ $\quad i=1,\ 2,\cdots\cdots,\ \mathrm{m}.$

(3) $x_j=0$ または 1 $\quad j=1,\ 2,\cdots\cdots,\ \mathrm{n}.$

（ここに b_i, a_{ij} はすべて実数，c_j はすべて非負の実数である）

集合 N および M を

$$N=\{1,\ 2,\cdots\cdots,\ \mathrm{n}\},$$
$$M=\{1,\ 2,\cdots\cdots,\ \mathrm{m}\},$$

と定義する．M の任意の元 i について，$a_{ij}<0$ であるような j の集合を N_i で表わす．すなわち，

(4) $N_i=\{j|j\in N$ かつ $a_{ij}<0\}$ $\quad i=1,\ 2,\cdots\cdots,\ \mathrm{m}.$

N_i を用いると，不等式（2）はつぎのように変形できる．

(5) $\sum_{j\in N_i^c} a_{ij}\ x_j+\sum_{j\in N_i}(-a_{ij})(1-x_j)\leqq b_i-\sum_{j\in N_i} a_{ij}$ $\quad i=1,\ 2,\cdots\cdots,\ \mathrm{m}.$

ここに N_i^c は N_i の N に関する補集合を表わす．いま，(5)式の右辺を新たに b_i, N_i に含まれる任意の j について $-a_{ij}$ を新たに a_{ij} とおき，$\bar{x}_j$ を

(6) $\bar{x}_j=1-x_j$

と定義すると，問題 IP. 1 はつぎの問題 IP. 2 に変換できる．

問題 IP. 2 (8)，(9)の制約のもとに(7)を最小にせよ．

(7) $z=\sum_{j\in N} c_i\ x_j.$

(8) $\sum_{j\in N_i^c} a_{ij}\ x_j+\sum_{j\in N_i} a_{ij}\ \bar{x}_j\leqq b_i$ $\quad \forall i\in M.$

(9) $x_j=0$ または 1 $\quad \forall\cdot\in N.$

（ここに c_j, a_{ij} はすべて非負である）

以下問題 IP. 2 の形で議論をする．

行列 A, B および列ベクトル $\boldsymbol{b}$ をつぎのように定義する．

(10) $A=(a_{ij})$ $\quad \boldsymbol{m}\times\boldsymbol{n}$ マトリックス

$B=(b_{ij})$ $\quad \boldsymbol{m}\times\boldsymbol{n}$ マトリックス

(11) $\boldsymbol{b}=(b_i)$ $\quad \boldsymbol{m}$ 次元列ベクトル

ここで，B の要素 b_{ij} はつぎのように定める．

(12) $b_{ij}=1 \quad \forall j\in N_i.$

$\quad\ =0 \quad \forall j\in N_i^c.$

A を係数マトリックス，$\boldsymbol{b}$ を定数項ベクトルとよぶ．

3. 解　系　列

0および1からなる集合をQとする．Qのn個の直積をQ^nと書く．Q^nをXで表わすことにする．Xの任意の元を解という．解をベクトルxあるいはnタップル$(x_1, x_2 \cdots\cdots, x_n)$で表わす．すると問題IP. 2は「条件(8)を満たす解で(7)を最小にするものを見いだせ」といいかえることができる．(8)を満たす解を許容解，問題IP. 2の解を最適解という．許容解の全体をFによって表わす．

つぎに，アルゴリズムの構成の上で必要となる解系列およびその他の諸定義をする．

Nの相異なる元からなる系列をインデックス列という．任意のインデックス列をIで表わす．インデックス列についてつぎの定義をする．$I=i_1\ i_2 \cdots\cdots i_l$とするとき，

(1)　$L(I)=i_l$,

(2)　$\varphi(I)=\{i_1, i_2, \cdots\cdots, i_l\}$,

(3)　$i \notin \varphi(I)$とするとき，$I \cdot i=i_1\ i_2 \cdots\cdots i_l\ i$,

(4)　$I/L(I)=i_1\ i_2 \cdots\cdots i_{l-1}$,

(5)　空系列をΛで表わす，

(6)　系列の長さを$lg(I)$で表わす．

また，Qの元からなる任意の有限系列をVによって表わす．いま$V=\alpha_1\ \alpha_2 \cdots\cdots \alpha_l$とするとき，インデックス列と同様につぎの定義をする．

(1)　$\alpha \in Q$について$V \cdot \alpha=\alpha_1\ \alpha_2 \cdots\cdots \alpha_l\ \alpha$,

(2)　$L(V)=\alpha_l$,

(3)　$V/L(V)=\alpha_1\ \alpha_2 \cdots\cdots \alpha_{l-1}$.

長さの等しい任意のVとIによって，解系列を

$$(V, I)$$

と定義する．解系列をSで表わし，解系列の全体（インデックス列の定義からV, Iともにその長さがn以下であることに注意）をΣで表わす．任意の解系列はXの一つの部分集合に対応している．その対応はつぎに定義する関数ϕ_jとΓを用いて表わされる．

いま，0，1およびシンボル$*$からなる集合をQ^*とするとき，Nのすべての元jについて，ΣからQ^*への関数ϕ_jをつぎのように定義する．$S=(\alpha_1\ \alpha_2 \cdots\cdots \alpha_l, j_1\ j_2 \cdots\cdots j_l)$とするとき，

$$\phi_{j_i}(S)=\alpha_i \qquad \forall i \in \{1, 2, \cdots\cdots, l\}, \tag{13}$$

$$\phi_j(S)=* \qquad \forall i \notin \{j_1, j_2, \cdots\cdots, j_l\}.$$

また，ΣからXの巾集合（$P(X)$などと書く）への関数Γを，任意のSについて，

$$\Gamma(S)=\{(x_1, x_2, \cdots\cdots, x_n) \mid \phi_j(S) \neq * \text{ のとき } x_j=\phi_j(S) \text{ かつ } \phi_j(S)=* \text{ のとき } x_j \text{ は0または1}\} \tag{14}$$

と定義する．すなわち，解系列SはΓによってXの部分集合の一つに写像される．

4. 解系列による解法

問題 IP. 2 の解法は，ある指定された順序で，x の要素に次々と 0 または 1 の値を割り当て，その割り当てが許容解となりうるか否かをたえず判別し，まだ x の要素で割り当てのすんでいないものがあっても許容解となりえないとわかったときバックトラックする[3]，という考え方にもとづくものである．ここで，ある段階まで割り当てのすんだ状態は解系列を用いて表わされる．したがって，ここでの判別とは，ある解系列 S について，X の部分集合 $\Gamma(S)$ が許容解を含むかどうかの判別を行なうということを意味する．「判別」は制限関数というものでおきかえることができる．まず制限関数の定義を述べる．

いま，S を任意の解系列とするとき，X の部分集合 $\Gamma(S)$ について $\Gamma(S)\cap F=\phi$ が成立するか否かの判別を行なう．Δ を $P(X)$ から Q への関数とするとき，$\Gamma(S)\cap F=\phi$ と判別できたなら

$$\Delta(\Gamma(S))=0,$$

判別できないとき，

$$\Delta(\Gamma(S))=1$$

と関数 Δ の値を定める．このような Δ を F- 制限関数という．一般に，F- 制限関数 Δ とは，つぎの(1)および(2)の条件を満足する $P(X)$ から Q への関数であると定義する．

(1) S の長さが n より小さいとき，$\Delta(\Gamma(S))=0$ ならば $\Gamma(S)\cap F=\phi$.

(2) S の長さが n のとき，$\Delta(\Gamma(S))=0$ となるのは $\Gamma(S)\cap F=\phi$ のとき，およびそのときのみに限る．

$\Gamma(S)$ なる形以外の X の部分集合については，Δ の値は任意でよい．制限関数の実際の構成例は 5 で示す．

いま不等式条件を

$$\text{(15)}\qquad \sum_{j\in N} c_j x_j \leqq \xi,$$
$$\sum_{i\in N_i^c} a_{ij}x_j + \sum_{j\in N_i} a_{ij}\bar{x}_j \leqq b_i \qquad \forall i\in M$$

と与える．ここで ξ は任意の実数である．

(15)を満たす解の全体を F_ξ と書く．任意の実数 ξ について，F_ξ は F に含まれる．とくに $\xi=\infty$ とするとき $F_\xi=F$ である．

F-制限関数の構成が可能であれば，同様にして F_ξ-制限関数の構成が可能である．このことは **5** での実際の制限関数の構成の仕方から明らかにされる．任意の実数 ξ について Δ_ξ を F_ξ-制限関数とするとき，問題 IP. 2 を解くアルゴリズムを構成すると，つぎのようになる．

IP. 2 を解くアルゴリズム

ステップ 1. $\xi=\infty$, $S=(V, I)=(\lambda, \lambda)$ とおく．ステップ 2 へ．

ステップ 2.　$S=(V, I)$ のとき，$j\notin\varphi(I)$ なる任意の j について

$$\Delta_\xi(\Gamma(V\cdot 0,\ I\cdot j))=1$$

ならば $\alpha=0$ とおいてステップ4へ．

$$\Delta_\xi(\Gamma(V\cdot 0,\ I\cdot j))=0$$

ならばステップ3へ．

ステップ 3.

$$\Delta_\xi(\Gamma(V\cdot 1,\ I\cdot j))=1$$

ならば $\alpha=1$ とおいてステップ4へ．

$$\Delta_\xi(\Gamma(V\cdot 1,\ I\cdot j))=0$$

ならばステップ5へ．

ステップ 4.　$lg(I\cdot j)=n$ のとき $S^*=(V\cdot\alpha,\ I\cdot j)$ を記憶し

$$S=(V/L(V),\ I/L(I)),$$

$$\xi=\sum_{j\in N}c_j\phi_j(S^*)-\varepsilon$$

とおいてステップ2へ．$lg(I\cdot j)<n$ のとき，

$$S=(V\cdot\alpha,\ I\cdot j)$$

とおいてステップ2へ．

ステップ 5.　$lg(I)=0$ のとき終了．$lg(I)\geqq 1$ のとき，$L(V)=0$ ならば

$$j=L(I),$$

$$S=(V/L(V),\ I/L(I))$$

とおいてステップ3へ．$L(V)=1$ のときステップ6へ．

ステップ 6.

$$S=(V/L(V),\ I/L(I))$$

とおいてステップ5へ．

ここに ε は，

(16)
$$0<\varepsilon<\min_{(x,\,y)\in G}\left|\sum_{j\in N}c_jx_j-\sum_{j\in N}c_jx_j\right|$$

$$\left(\text{ただし，}G=\{(x,\ y)\mid\sum_{j\in N}c_jx_j\neq\sum_{j\in N}c_jy_j\}\right)$$

を満足する任意の数とする．もし，すべての c_j が整数であれば，　$\varepsilon=1$ とおくとき ε が(16)を満足することは明らかである．

上述のアルゴリズムにおいて，ステップ4で記憶された解系列 S^* はすべて F の元に対応しており，その記憶の順序にしたがって，$\sum_{j\in N}c_j\cdot\phi_j(S^*)$ の値が減少する．最も新しく記憶された S^* について，$\Gamma(S^*)$ が IP. 2 の最適解であることは，アルゴリズムの構成の仕方から明らかである．

ステップ2において，$\varphi(I)$ に含まれない N の一つの元を選択する仕方については種々の方法が考えられる．たとえば，あらかじめ N の元の選択の順序をきめる．すなわち，$1\rightarrow 2\rightarrow\cdots$

$\to n$ あるいはその順列の一つにとる．このような場合については，**6** で実際の例を用いて示す．

5. 制 限 関 数

4 で示したアルゴリズムにおいて，制限関数は最も重要な役割を果しており，アルゴリズムの良さは制限関数をいかに定めるかということに依存している．ここでは，制限関数を具体的にどのように構成するかを示す．（8）の不等式系を満たす解の全体をこれまでと同様に F とする．F-制限関数の例として，つぎに述べる（1），（2），（3）の三つが考えられる．

（1） 任意の解系列 S について，$\boldsymbol{m}$ 次元列ベクトル $\boldsymbol{b}(S)$ をつぎのよとに定義する．$\boldsymbol{b}(S)$ の $\boldsymbol{i}$ 要素 $b_i(S)$ を

$$b_i(S)=b_i-\Big(\sum_{j\in E_i^{(1)}} a_{ij}\phi_j(S)+\sum_{j\in E_i^{(2)}} a_{ij}\bar{\phi}_j(S)\Big). \tag{17}$$

（ここに，$E_i^{(1)}=N_i^c\cap\{j|\phi_j(S)\neq *\}$,

$E_i^{(2)}=N_i\cap\{j/\phi_j(S)\neq *\}$,

また，　$\bar{\phi}_j(S)=1-\phi_j(S)$）

任意の解系列 S について，$\Delta(\Gamma(S))$ を $\boldsymbol{b}(S)$ の要素で $b_i(S)<0$ となるものが存在するとき，およびそのときに限り 0 とおく．

このような Δ について，$S=(\mathrm{V}, I)$ とするとき，つぎのことがいえる．

（i） $lg(I)<n$ なら，$\Delta(\Gamma(S))=0$ のとき $\Gamma(S)\cap F=\phi$.

（ii） $lg(I)=n$ なら，$\Delta(\Gamma(S))=0$ と $\Gamma(S)\cap F=\phi$ とは同値である．

このことは a_{ij} がすべて非負ということから明らかである．すなわち Δ は F-制限関数となっている．このような Δ を $\Delta^{(1)}$ で表わしておく．

（2） $\boldsymbol{x}^*=(x_1^*, x_2^*,\cdots\cdots, x_n^*)$ を任意の $\boldsymbol{M}$ の元について，

$$x_j^*=b_{tj} \qquad \forall j\in N \tag{18}$$

とおく．（8）式の $i=t$ の左辺を計算すると，

(12)式より

$$\sum_{j\in N_t^c} a_{tj}x_j^*+\sum_{j\in N_t} a_{tj}\bar{x}_j^*=0 \tag{19}$$

である．いま $\boldsymbol{m}\times\boldsymbol{n}$ マトリックス $\boldsymbol{A}^{(t)}=(a'_{ij})$ を

$$a'_{ij}=(b_{tj}\dot{+}b_{ij})\times a_{ij} \qquad \forall i\in M,\ \forall j\in N \tag{20}$$

とおく．ここに，$\dot{+}$ は 2 を法とする加法を表わす*1．N のある部分集合 E について

$$\sum_{j\in E} a'_{ij}>b_i,$$

かつすべての E の元 j' について

$$\sum_{j\in E-\{j'\}} a'_{ij}\leqq b_i$$

を満足する $\boldsymbol{A}^{(t)}$ の行 i が存在するとき，

$$x_j' = b_{lj} \qquad \forall j \in E, \tag{21}$$
$$= 0 \text{ または } 1 \qquad \forall j \notin E$$

なる解 $\boldsymbol{x}' = (x_1', x_2', \cdots\cdots, x_n')$ は F の元とはなり得ない．なぜなら

$$\begin{aligned} &\sum_{j\in N_i^c} a_{ij}x_j' + \sum_{j\in N_i} a_{ij}\bar{x}_j' \\ &\geqq \sum_{j\in N_i^c\cap E} a_{ij}x_j' + \sum_{j\in N_i\cap E} a_{ij}\bar{x}_j' \\ &= \sum_{j\in N_i^c\cap E} a_{ij}(b_{lj}\dot{+}0) + \sum_{j\in N_i\cap E} a_{ij}(b_{lj}\dot{+}1) \\ &= \sum_{j\in E} a_{ij}' > b_i. \end{aligned} \tag{22}$$

したがって，このような E が存在するとき，E に含まれる少なくとも一つの元 j について

$$x_j \neq b_{lj}$$

であるような解 $\boldsymbol{x}$ でなければ F の元にはなり得ない．E を $\boldsymbol{x}^*$ に対する E 集合という．

いま，$E_1, E_2, \cdots\cdots E_l$ が $\boldsymbol{x}^*$((18)で与えた）に対する $\hat{E}$ 集合でたがいに共通部分を持たないとするとき，上で与えた説明により $\boldsymbol{x}$ が F の元であるためには，E_i に含まれる少なくとも一つの j について

$$x_j \neq b_{lj}$$

でなければならないから，F の任意の元 $\boldsymbol{x}$ について

$$\sum_{j\in N_l^c} a_{lj}x_j + \sum_{j\in N_l} a_{lj}\bar{x}_j \geqq \sum_{i=1}^{l} \min_{j\in E_i} a_{lj} \tag{23}$$

が成立する．したがって

$$\sum_{i=1}^{l} \min_{j\in E_i} a_{lj} > b_l \tag{24}$$

であれば，$F=\phi$ であることがいえる．

たがいに共通部分を持たない(18)で与えた $\boldsymbol{x}^*$ に対する E 集合の求め方を次に示そう．

$\boldsymbol{A}^{(l)}$ の j 番目の列ベクトルを $A_j^{(l)}$ で表わす．また同次元の列ベクトル $\boldsymbol{a}, \boldsymbol{b}$ について，その要素ごとに大小を比較したとき少なくとも一つの i について $a_i > b_i$ なら

$$\boldsymbol{a} > \boldsymbol{b}$$

と書く．$\boldsymbol{a}\pm\boldsymbol{b}$ は要素ごとの加減算を表わす．c を任意の実数とするとき $c\cdot\boldsymbol{a}$ は $\boldsymbol{a}$ の各要素に c をかけることを意味する．

E 集合を求めるアルゴリズム

ステップ 1．すべての $j(\in N)$ について $c_j=1$ とおく．また $k=0$, $\omega^{(0)}=0$ とおいてステップ 2へ

ステップ 2

*1 ここに記号$\dot{+}$は，$j\in N_i^c$ ならば $a'_{ij}=b_{lj}\cdot a_{ij}$, $j\in N_i$ ならば $a'_{ij}=\bar{b}_{lj}\cdot a_{ij}$ ということを表わすため用いた．

$$\sum_{j \in N} c_j \cdot A_j^{(t)} > \boldsymbol{b}$$

なら

$$E = \{j / c_j = 1\}$$

とおいて，ステップ3へ．上のことが成立しないとき終了．

ステップ 3. E に含まれる i について

$$\sum_{j \in E - \{i\}} c_j \cdot A_j^{(t)} > \boldsymbol{b}$$

が成立するとき，ステップ4へ．どのような E の元 i についてもこのことが成立しないときステップ5へ．

ステップ 4. $E = E - \{i\}$ とおいてステップ3へ．

ステップ 5. E を E_k とおいてこれを記憶する．

$$\omega^{(k+1)} = \omega^{(k)} + \min_{j \in E} a_{tj},$$

$$c_j = 0 \qquad \forall j \in E_0 \cup E_1 \cup \cdots\cdots \cup E_k,$$

$$= 1 \qquad \forall j \notin E_0 \cup E_1 \cup \cdots\cdots \cup E_k$$

とおき，$k = k+1$ としてステップ2へ．

このようなアルゴリズムによって最終的に得られた $\omega^{(k)}$ を ω_t とするとき，F に含まれる，任意の元 $\boldsymbol{x}$ について

$$\sum_{j \in N_t^c} a_{tj} x_j + \sum_{j \in N_t} a_{tj} \bar{x}_j \geqq \omega_t$$

が成立することは明らかである．アルゴリズムで得られた E_1, E_2,……はたがいに共通部分を持たない E 集合となっている．

〔例題〕不等式系，

$$5x_1 + 8x_2 + 7x_3 + 3x_4 + 9x_5 + 2x_6 + 4x_7 \leqq 19, \tag{25}$$

$$8\bar{x}_1 + 6\bar{x}_2 + 7x_3 + 6x_4 + 2\bar{x}_5 + 3x_6 + 7\bar{x}_7 \leqq 5,$$

$$5\bar{x}_1 + 4\bar{x}_2 + x_3 + 9\bar{x}_4 + 7\bar{x}_5 + 2x_6 + 2x_7 \leqq 15$$

について，(0, 0, 0, 0, 0, 0, 0,) に対する E 集合を求めてみよう．まず，$\boldsymbol{A}^{(1)}$ は

$$\boldsymbol{A}^{(1)} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 6 & 0 & 0 & 2 & 0 & 7 \\ 5 & 4 & 0 & 9 & 7 & 0 & 0 \end{vmatrix} \tag{26}$$

となる．E 集合を求めるアルゴリズムによって

$$E_1 = \{1\}, \quad E_2 = \{2\}, \quad E_3 = \{2\}, \quad E_3 = \{4, 5\}, \quad E_4 = \{7\}$$

を得る．$\omega_1 = 20$ である．ω_1 の値が19より大であることは不等式(25)を満足する解が存在しないことを意味する．

$t = 1, 2, \cdots\cdots, m$ のそれぞれの場合について，ω_t を求め，列ベクトル (ω_t) を ω と書く．ω と定数項ベクトル $\boldsymbol{b}$ について，

(27) $$\boldsymbol{w}>\boldsymbol{b}$$

が成立するとき，F は空である．

以上のことを制限関数の構成に用いることができる．任意の解系列 S について，不等式系を

(28) $$\sum_{j\in N_i^{(1)}} a_{ij}x_j+\sum_{j\in N_i^{(2)}} a_{ij}\bar{x}_j\leqq b_i(S) \qquad \forall i\in M$$

（ここに，$N_i^{(1)}=N_i^c\cap\{j/\phi_j(S)=*\}$,

$N_i^{(2)}=N_i\cap\{j/\phi_j(S)=*\}$）

とおく．不等式系(8)に対して行なったと同様にして，$\omega_1, \omega_2, \cdots\cdots, \omega_m$ を求める．これは S に依存しているから，それぞれを $\omega_i(S)$, その列ベクトルを $\omega(S)$ によって表わすことにする．したがって

(29) $$\omega(S)>\boldsymbol{b}(S)$$

が成立するとき，(28)を満足する解は存在しない．X の巾集合から Q への関数 Δ の $\Gamma(S)$ の値を

$$\omega(S)>\boldsymbol{b}(S)$$

が成り立つとき，およびそのときに限り

$$\Delta(\Gamma(S))=0$$

とおく．このような Δ が F-制限関数であることは明らかである．ここでの Δ を $\Delta^{(2)}$ と書く．

（3） 2） 任意の解系列 S について，N の部分集合 $E_0(S)$, $E_1(S)$ をつぎのようにおく．

(30) $E_0(S)=\{i|$ 少なくとも一つの $i(\in M)$ について，$j\in N_i^{(1)}$かつ $a_{ij}>b_i(S)\}$,

$E_1(S)=\{j|$ 少なくとも一つの $i(\in M)$ について，$j\in N_i^{(2)}$かつ $a_{ij}>b_i(S)\}$.

（ここに，$N_i^{(1)}=N_i^c\cap\{j|\phi_j(S)=*\}$,

$N_i^{(2)}=N_i\cap\{j|\phi_j(S)=*\}$）

このような $E_0(S)$, $E_1(S)$について

(31) $$E_0(S)\cap E_1(S)\neq\phi$$

が成立するとき，$\Gamma(S)\cap F=\phi$ であることは明らかである．したがって，$\Delta(\Gamma(S))$ を

$$\boldsymbol{b}(S)<\boldsymbol{0}$$

（ここに $\boldsymbol{0}$ は $\boldsymbol{m}$ 次元零ベクトル）

あるいは

$$E_0(S)\cap E_1(S)\neq\phi$$

が成立するとき，およびそのときに限り

$$\Delta(\Gamma(S))=0$$

とおく．Δ が制限関数であることは明らかである．このような Δ を $\Delta^{(3)}$ と書く．

（1），（2），（3）で与えた制限関数 $\Delta^{(1)}$, $\Delta^{(2)}$ および $\Delta^{(3)}$ について，S を任意の解系列とするとき

(32) $$\Delta^{(1)}(\Gamma(S)) \geqslant \Delta^{(3)}(\Gamma(S)) \geqslant \Delta^{(2)}(\Gamma(S))$$

が成立する．

不等式系(15)に対する制限関数 $\Delta^{(1)}$, $\Delta^{(2)}$ $\Delta^{(3)}$ が構成されることは，これまでの議論から明らかである．任意の実数 ξ について，不等式系(15)に対する制限関数 $\Delta^{(1)}$, $\Delta^{(2)}$, $\Delta^{(3)}$ を $\Delta^{(1)}{}_{\xi}$, $\Delta^{(2)}{}_{\xi}$, $\Delta_{\xi}{}^{(3)}$ と書く．

6, 例　　　題

この節では制限関数 $\Delta_{\xi}^{(1)}$による解法例を示す．$\hat{\boldsymbol{b}}_{\xi}(S)=(\hat{b}_i(S))$ を$m+1$ 次元ベクトルとし，

(33) $$\hat{b}_i(S)=b_i(S) \qquad 1\leqslant i\leqslant m,$$

$$\hat{b}_{m+1}(S)=\xi-\sum_{j\in\{j\ \varphi_j(S)\neq *\}} c_j\phi_j(S)$$

と定義する．

〔例題1〕[1]

(34) $$Z=5x_1+7x_2+10x_3+3x_4+x_5.$$

(35) $$-x_1+3x_2-5x_3-x_4+4x_5\leqslant-2,\ 2x_1-6x_2+3x_3+2x_4-2x_5\leqslant 0,$$

$$x_2-2x_3+x_4+x_5\leqslant-1.$$

この問題を IP 2. に変換すると，係数マトリックス $\boldsymbol{A}$，定数項ベクトル $\boldsymbol{b}$ およびマトリックス $\boldsymbol{B}$ はつぎのようになる．

$$\boldsymbol{A}=\begin{vmatrix}1&3&5&1&4\\2&6&3&2&2\\0&1&2&1&1\end{vmatrix} \qquad \boldsymbol{b}=\begin{vmatrix}5\\8\\1\end{vmatrix}$$

$$\boldsymbol{B}=\begin{matrix}1&0&1&1&0\\0&1&0&0&1\\0&0&1&0&0\end{matrix}$$

新たな N の元の選択を $1\to2\to3\to4\to5$ の順序で行なう．したがって，以下解系列 $S=(I,V)$ をそのインデックス列の項を省略して，$S=V$ と略記する．(16)の ε の値は，上のような問題の場合，$\varepsilon=1$ とおけばよい．

（1） $\xi=\infty$, $S=\lambda$(ステップ 1.).

（2）*2 $\hat{\boldsymbol{b}}_{\infty}(0)=(4,8,1,\infty)$, したがって $\Delta_{\infty}^{(1)}(0)=1$(ステップ2)．$S=0$ となる（ステップ4）.

(3) $\hat{\boldsymbol{b}}_\infty(00)=(4,\ 2,\ 1,\ \infty)$, したがって $\Delta_\infty^{(1)}(00)=1$ (ステップ 2). $S=00$(ステップ 4).

(4) $\hat{\boldsymbol{b}}_\infty(000)=(-1,\ -2,\ -1,\ \infty)$, したがって $\Delta_\infty^{(1)}(000)=0$ (ステップ 2). $\hat{\boldsymbol{b}}_\infty(001)=(4,\ -1,\ 1,\ \infty)$, したがって $\Delta_\infty^{(1)}(001)=0$ (ステップ 3). つぎに $S=01$ とする (ステップ 5). $\hat{\boldsymbol{b}}_\infty(01)=(1,\ 8,\ 0,\ \infty)$, したがって $\Delta_\infty^{(1)}(01)=1$ (ステップ 3). $S=01$ となる (ステップ 4).

(5) $\hat{\boldsymbol{b}}_\infty(010)=(-4,\ 8,\ -2,\ \infty)$, したがって $\Delta_\infty^{(1)}(010)=1$ (ステップ 2). $\hat{\boldsymbol{b}}_\infty(011)=(1,\ 5,\ 0,\ \infty)$, したがって $\Delta_\infty^{(1)}(011)=1$ となり (ステップ 3), $S=011$ (ステップ 4).

(6) $\hat{\boldsymbol{b}}_\infty(0110)=(0,\ 5,\ 0,\ \infty)$, したがって $\Delta_\infty^{(1)}(0110)=1$ (ステップ 2). $S=0110$ (ステップ 4).

(7) $\hat{\boldsymbol{b}}_\infty(01100)=(0,\ 3,\ 0,\ \infty)$, したがって $\Delta_\infty^{(1)}(01100)=1$ (ステップ 2). S の長さは 5 であるから, $(0,\ 1,\ 1,\ 0,\ 0)$ は $\boldsymbol{F}_\infty$ の元である (ステップ 4). そこで, $\xi=17-1=16$, $S=0110$ とおいてステップ 2 へ.

(8) 以下同様にして $F_{16}=\phi$ であることがわかる. 以上の計算によって, 解 (0, 1, 1, 0, 0,) が問題の最適解であることが示された. そのときの Z の値は17である.

〔例題 2〕(2)

$$Z=3x_1+2x_2+5x_3+8x_4+6x_5+0x_6+11x_7+4x_8+5x_9+6x_{10}+11x_{11}+2x_{12}+8x_{13}+5x_{14}+8x_{15}+7x_{16}+3x_{17}+9x_{18}+2x_{19}+4x_{20}. \tag{36}$$

また, 係数マトリックス $\boldsymbol{A}$, 定数項ベクトル $\boldsymbol{b}$ およびマトリックス $\boldsymbol{B}$ がつぎのように与えられているとする.

$$\boldsymbol{A}=\begin{vmatrix} 6&5&8&3&0&1&3&8&9&3&8&6&3&8&6&7&6&2&3&7 \\ 1&3&3&4&1&0&4&1&6&0&8&0&1&5&4&1&9&7&2&2 \\ 3&6&1&5&5&6&9&6&3&9&6&3&6&6&6&2&7&6&0&7 \end{vmatrix}$$

$$\boldsymbol{b}=\begin{vmatrix} 51 \\ 24 \\ 53 \end{vmatrix}$$

$$\boldsymbol{B}=\begin{vmatrix} 1&0&1&1&0&1&1&1&1&0&1&0&1&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&1&1&0&0&0&0&1&1&1&1&1&0&0 \\ 0&0&0&1&1&0&1&0&0&1&1&1&1&1&0&0&1&1&0&1 \end{vmatrix}$$

N の元の選択の順序を $1\to2\to3\to\cdots\cdots\to20$ とするときの計算結果をつぎに示す. 例題 1 と同様に, $\varepsilon=1$ とおけばよい.

(1) $\xi=\infty$ のとき

$$S=00000000000011100101$$

を得る. すなわち, $\boldsymbol{F}_\infty(=\boldsymbol{F})$ の元

*2 以下, $\Delta_\xi^{(1)}(\Gamma(S))$ を $\Delta_\xi^{(1)}(S)$ と略記する.

$$(0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 1,\ 0,\ 1,)$$

を得る．その Z の値は34である．

(2) $\xi=33$ のとき，

$$S=00000000100001000101$$

を得る．$Z=23$.

(3) $\xi=22$ のとき，

$$S=00100000100001001001$$

を得る．$Z=22$.

(4) $\xi=21$ のとき，

$$S=10000000000001000101$$

を得る．$Z=21$.

(5) $\xi=20$ のとき，

$$S=10000000100001001001$$

を得る．$Z=20$.

(6) $\xi=19$ のとき，$F_{19}=\phi$ である．

以上の計算から，最適解は

$$(1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 1)$$

となり，その Z の値は20である．

Nの元の選択の順序を $1\to2\to3\to\cdots\to20$のかわりに，$3\to4\to5\to\cdots\to20\to1\to2$ とすると，まず $Z=21$ になる解が得られ，つぎに $Z=20$ になる解（最適解）を得る．前者の HITAC 5020 E による計算時間は1秒強，後者の場合，1秒以下である．すなわち，N の元の選択順序は計算時間を左右する．

7. むすび

以上，バックトラックという原理による 0—1 整数計画問題の解法を示した．バックトラックなる名称は D.H. Lehmer によるものであるが，名称が与えられる以前から種々の問題の解法において用いられた原理である．それは本文の解法からも明らかなように非常に簡単な原理である．本論文のアルゴリズムの効力は制限関数のよさ（計算が簡単で，X の多くの部分集合 Y について，それを $\Delta(Y)=0$ という理由により除外できること）に依存している．制限関数については(1)，(2)および(3)の三つの場合をあげたがそれ以外にも考えられよう．しかし，その計算があまり複雑になると，かえってアルゴリズムの効力が低下してしまうということに注意しなければならない．単に計算の複雑さということを比較すると，$\Delta^{(1)}$ より $\Delta^{(3)}$，$\Delta^{(3)}$ より $\Delta^{(2)}$ がより複雑になっている．**6** 節の例題 1，2 のような次元の低い問題に対して，$\Delta^{(2)}$ あるいは $\Delta^{(3)}$ を用い

ることは適当ではない．しかし，より高次元の問題に対し，$\Delta^{(2)}$ あるいは $\Delta^{(3)}$（とくに $\Delta^{(2)}$）がどの程度の効力を有するかということは未知であり，今後検討すべき課題の一つである．

最後に，常に御指導を賜わる本多波雄教授，例題を解くプログラムについて御協力下さった星光治氏，ならびに御討論いただいた本多研究室の方々に深謝いたします．

参考文献

1 E. Balas, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," Opns. Res. 13, 517-546 (1965).

2. E. Balas, "Discrete Programming by the Filter Method," Opns. Res. 15, 915-956 (1967).

3. Solomon W. Golomb and Leonald D. Baumert, "Backtrack Programming." J.A.C.M. 12, No. 4. 516-524 (1965).

4. 阿部健一，木村正行，"離散型決定問題についての一考察"，日本オペレーションズ・リサーチ学会秋季研究発表会アブストラクト集11～12 (1968).

5. 阿部健一他，"Integer Programming の組み合わせ論的解法"，オートマトンと自動制御研究会資料(1966年9月).

6. G.B. Dantzig, *Linear Programming and Extensions*, Princeton Univ. Press.