



HOKKAIDO UNIVERSITY

Title	凸多面体の端点の階級付けアルゴリズム
Author(s)	大内, 東; Oouchi, Azuma; 加地, 郁夫 他
Citation	北海道大學工学部研究報告, 67, 55-62
Issue Date	1973-06-30
Doc URL	https://hdl.handle.net/2115/41135
Type	departmental bulletin paper
File Information	67_55-62.pdf



凸多面体の端点の階級付けアルゴリズム

大内 東* 加地 郁夫*

(昭和47年11月30日受理)

An Algorithm for Ranking All Extreme Points

Azuma OOUCHI and Ikuo KAJI

(Received November 30, 1972)

Abstract

An algorithm for searching and leveling all extreme points of a convex polyhedron corresponding to a linear programming problem in a decreasing order of the linear objective function is described.

1. ま え が き

最適計画問題に於いて、「最適解が実行可能解の集合の端点 (EXTREME POINT, EP) 上に於いて起る」という問題は数多い。従ってこれらの端点を調らべる必要がしばしば起る。本報告ではこれらの端点の階級付け (すべての端点を目的関数の値の大きい順に並べること) の一方法について述べたものである。ここに述べたアルゴリズムの対象としている最適問題は LP であり、また縮退は起らないと仮定している。

本報告の概要は次のようになっている。第2章では、アルゴリズムの基となる凸多面体の重要な性質について、これを定理として表わし、証明を行なっている。第3章では、第2章の定理に基いて考案したアルゴリズムについて述べ、更にこのアルゴリズムの実行される様子を二つの例題を使って説明している。

2. 理論的背景

次の問題“LP(I)”と“S(I)”を設定する。

LP(I)

$$\begin{aligned} \max z &= \mathbf{c}\mathbf{x} \\ \text{sub. to;} \quad \mathbf{A}\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

S(I)

LP(I) の実行可能解の端点を階級付けせよ。

ここで、 \mathbf{A} は $m \times n$ の実数のマトリクス、 \mathbf{b} は m 次元空間、 E^m の m 成分ベクトル、 \mathbf{c} と \mathbf{x} は E^n の n 成分ベクトル

である。

以下において、LP(I) は解を持ち、縮退は起らないとして議論を進める。

* 電気工学科

2.1 定 義

次の事柄を定義する¹⁾。(第1図, 第2図)

定義1. 辺 (EDGE)

x_1^* , x_2^* を凸集合 X 内の異なる任意の二つの端点とする。これらの二点を結ぶ線分が X と支持超平面との“まじわり”であるとき、この線分を凸集合 X の辺であると云う。(第1図では、例えば AB は辺である)

また x_1^* を X の一つの端点とし、 X 内の他の点 x を $x = x_1^* + \lambda(x - x_1^*)$ がすべての $\lambda \geq 0$ に対して X 内にあるような点とする。集合 $L = \{x | x = x_1^* + \lambda(x - x_1^*), \text{ all } \lambda \geq 0\}$ が X と支持超平面との“まじわり”であるとき、 L は無限に伸びる X の辺であると云う。

定義2. 隣接する端点 (ADJACENT EP)

凸集合の異なる任意の二つの端点 x_1^* と x_2^* はこれらを結ぶ線分が凸集合の辺であるとき隣接していると云う。(第1図では、 B と C とは A に隣接している端点である)

定義3. 端点のレベル (LEVEL of EP)

目的関数の値が $z_1 > z_2$ であるような二つの端点 x_1^* と x_2^* に対して、 x_1^* は x_2^* よりレベルが上にあると云い、逆に x_2^* は x_1^* よりレベルが下にあると云う。 $z_1 = z_2$ である場合は、 x_1^* と x_2^* は同じレベルにある、または x_1^* と x_2^* はそれぞれ他の alternative であると云う。(第1図に於いては、 A は第1レベルの端点、 C は第2レベルの端点、... etc.)

定義4. 凸多面錐 (CONVEX POLYHEDRAL CONE)

凸多面錐 C は有限個の半直線の和で定義される。i.e., $C = \sum L_i$ 。ここで和の意味は集合 A, B に対してその和 S は $S = \{a + b | a \in A, b \in B\}$ で定義する。(第2図では半直線 L_1, L_2 によって作られる凸多面錐を斜線で示してある)

定義5. EXTREME SUPPORTING HALF-SPACE

E^n の点集合 $H_F = \{v | cv \geq 0\}$ は、点 a_1, \dots, a_r ($r \geq n$) から生成される n 次元凸多面錐 C が半空間 H_F 内にあり、且つ集合 $\{a_1, \dots, a_r\}$ のうちの $(n-1)$ 個の線型独立な点が超平面 $cv = 0$ の上にあるとき、 C に対する“EXTREME SUPPORTING HALF-SPACE”と云う。(第2図における S_1)

定義6. EXTREME SUPPORTING HYPERPLANE

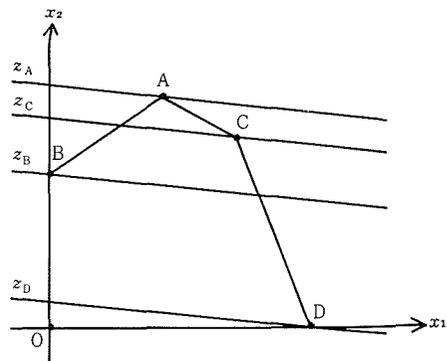
EXTREME SUPPORTING HALF-SPACE の境界を形成する超平面 $cv = 0$ は凸多面錐 C の“EXTREME SUPPORTING HYPERPLANE”である。(第2図に於ける π はその例である)

以上の定義のもとに次の補助定理を証明なしで述べる。

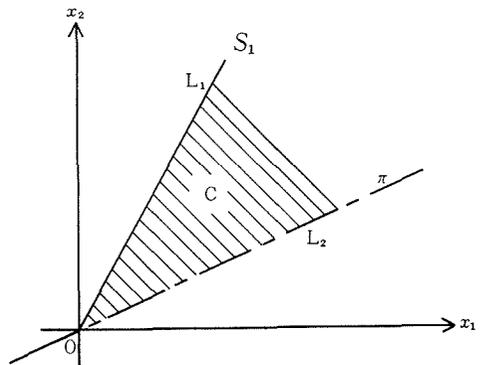
2.2 補 助 定 理

補助定理1.

r 個の点 a_1, \dots, a_r によって生成される n 次元凸多面錐 C に対して、すべての EXTREME



第1図 辺, 隣接する端点, 端点のレベルの例

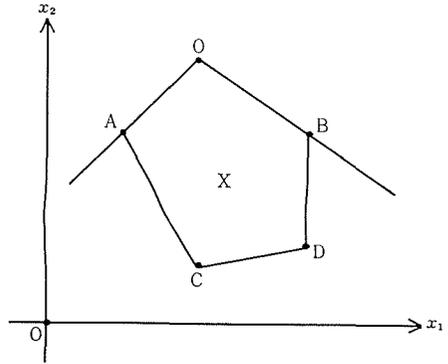


第2図 凸多面錐 C と extreme supporting hyperplane π

SUPPORTING HYPERPLANE π_i によって出来る EXTREME SUPPORTING HALF-SPACE H_{F_i} の“まじわり”は C 自身である。

補助定理 2.

n 次元凸多面体 X の任意の一つの端点を o とすれば, o と隣接しているすべての端点は o を頂点とする n 次元凸多面錐 C を生成する。この n 次元凸多面錐 C は X のすべての端点を含む。(第3図)



第3図 補助定理2の例

2.3 定 理

LP(I) の制約式によって定められる実行可能解の集合は凸多面体であり, 実行可能基底解がこの凸多面体の端点に対応し, 最適解はこれらの端点のうちの一つあるいは二つ以上で起ると言うことは周知である。このとき次の定理が成り立つ。

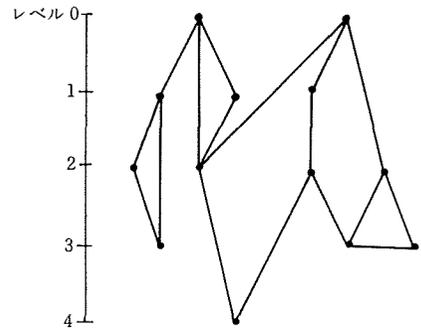
定理 1.

LP(I) の制約式を満足する端点の集合は連結である。i.e., 集合内の任意の二点は隣接する辺によって作られる道により結ぶことが出来る。

証明略す。

定理 2.

LP(I) に於いて, 最適値 $\max z = z_0$ を与える最適端点の集合 X_0 (i.e., 第0レベルの端点集合) と次善の値 $z = z_1, (z_0 > z_1)$ を与える第二最適端点の集合 X_1 (i.e., 第1レベルの端点集合) を考えると, X_1 に属する任意の端点に対して, これに隣接する端点が少なくとも一つは X_0 内に存在する。(第4図)



第4図 5レベルに分類される端点の隣接関係

証明

証明は背理法による。

\mathbf{x}_1^* を X_1 に属する任意の端点とする。 X_0 に含まれるいかなる端点も \mathbf{x}_1^* とは隣接していないと仮定する。 \mathbf{x}_1^* は目的関数から作られる超平面 $\pi_s; \mathbf{c}\mathbf{x} = \mathbf{c}\mathbf{x}_1$ 上にある。超平面 π_s は E^n を互に素である二つの半空間 S_1 と S_2 に分割する。すなわち,

$$S_1 = \{\mathbf{x} | \mathbf{c}\mathbf{x} > \mathbf{c}\mathbf{x}_1\}$$

$$S_2 = \{\mathbf{x} | \mathbf{c}\mathbf{x} \leq \mathbf{c}\mathbf{x}_1\}$$

$$S_1 \cap S_2 = \emptyset$$

$$S_1 \cup S_2 = E^n$$

仮定により半空間 S_1 内には集合 X_0 以外の他のどんな端点も存在しない。従って \mathbf{x}_1^* を頂点とし \mathbf{x}_1^* と隣接しているすべての端点より作られる凸多面錐 C は半空間 S_2 内にある。よって C は X_0 を含まない。これは補助定理2が述べている「 C が X のすべての端点を含む」という事実と反する。従って X_0 には \mathbf{x}_1^* と隣接している端点が少なくとも一つ存在する。証明終り。

定理 3.

LP(I) に於いて, ある z の値 $z = z_k$ を与える端点の集合 X_k (すなわち, 第 k レベルの端点集合) とそれよりレベルが上の端点の集合 $X_i, i = 0, 1, \dots, k-1$ を考えれば, X_k に属する任意の端点に対して, これに隣接する端点が少なくとも一つは $\{X_0 \cup X_1 \cup \dots \cup X_{k-1}\}$ 内に存在す

る。(第4図)

証明

定理2を帰納的に用いることにより証明できるので略す。

2.4 ALTERNATIVE OPTIMA についての考察²⁾

LP(I)の基底マトリクスを B , x_{Bi} をベクトル $x_B = B^{-1}b$ の第 i 成分とし, y_{ij} を $y_j = B^{-1}a_j$ の第 i 成分とする。このとき, シンプレックス法により得られる目的関数 z の増減は基底から出るベクトルを a_r , 新しく基底に入るベクトルを a_j とし, z の新しい値を \hat{z} で表わせば

$$\hat{z} = z + x_{Br}/y_{rj}(c_j - z_j) = z + \theta(c_j - z_j) \quad (2.4.1)$$

となる。すなわち, z の変化は $\theta(c_j - z_j)$ により決まる。従って最適値と同じ z の値を与える端点を得るためには, 最適解を与えている最終タブローにおいて, (a) $z_j - c_j = 0$ (j は非基底) のベクトル a_j を新しく基底に入れる, (b) $\theta = 0$, i.e., $x_{Br} = 0$ のベクトル b_r を基底から追い出す, のいずれかを行う事により新しい端点を求めれば, この端点は ALTERNATIVE となる。

2.5 ある端点と隣接している実行可能な端点を求める事

ある端点と隣接している端点を求める事は基底を一つ交換することにより得られる。掃き出しのピボットを (r, j) とすれば新しい基底変数の値 \hat{x}_{Bi} は次の式により得られる。

$$\hat{x}_{Bi} = x_{Bi} - y_{ij}x_{Br}/y_{rj}, \quad i \neq r \quad (2.5.1)$$

$$\hat{x}_{Bi} = x_{Br}/y_{rj} \quad (2.5.2)$$

ゆえに $\hat{x}_{Bi} \geq 0$ であるためには (2.5.2) より

$$y_{rj} \geq 0 \quad (2.5.3)$$

でなければならない。

また $y_{rj} > 0$, $y_{ij} \leq 0$ のときは (2.5.1) は自動的に満たされる。

$y_{ij} > 0$ のときは

$$x_{Bi}/y_{ij} - x_{Br}/y_{rj} \geq 0 \quad (2.5.4)$$

(2.5.3), (2.5.4) より $\hat{x}_{Bi} \geq 0$ であるためには, 次のようにして基底から出るベクトル a_r を決定すれば良い。

$$x_{Br}/y_{rj} = \min_i \{x_{Bi}/y_{ij}, y_{ij} > 0\} = \theta \geq 0$$

一方, 目的関数の増減は (2.4.1) 式で与えられるから z の増大のときは $z_j - c_j < 0$ である j を, 減少のときは $z_j - c_j > 0$ である j を選らば, 新しく基底に入るベクトルは決定される。

3. 問題 S(I) の解法アルゴリズム

前章の2.3で述べた定理に基いて S(I) の解法アルゴリズムが考えられる。3.1でこのアルゴリズムの概要を, 3.2で実際に用いられるラベリング技法について, それぞれ述べることにする。そのまえに以下の記号を定義しておく。

X ; $X = \{x^* | x^*$ は LP(I) の実行可能基底解}

x_0^* ; LP(I) の最適解

x_j^* ; LP(I) における第 i レベルで j 番目の端点

X_k ; $X_k = \{x_k^*\}$, for all j ($k=1, 2, \dots$, で, 特に X_1 は LP(I) の最適解の集合を表わす)

s_j^* ; X_i と隣接している第 j 番目の端点

S_k ; $S_k = \{s_k^*\}$, for all j ($k=1, 2, \dots$,)

J_i ; $J_i = \{j | j$ は x^* の基底を構成しているベクトル a_j の添字 $j\}$

J ; $J = \{\text{あるステップまでに求めたすべての } J_i\}$
 $-$; $A - B$ で集合 A と集合 B の差集合を表わす。

3.1 アルゴリズムの概要

アルゴリズムの概要は以下の様である。

(1) シンプレックス法により LP(I) の最適解 \mathbf{x}_0^* を求める。

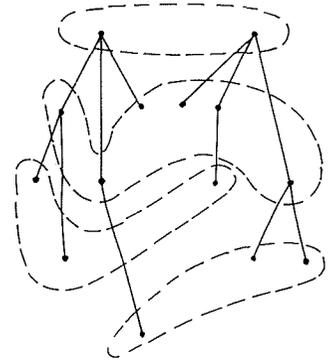
(2) \mathbf{x}_0^* の ALTERNATIVE を求める。これで X_1 が決定出来た。(但し, $J_i \in J$ の解は改らためて求める必要はない)

以下は繰り返えしの手順となる。

(3) X_k と隣接しているすべての端点を求める。 S_k とする。($J_i \in J$ は除く)

(4) 集合 $\{(S_1 \cup S_2 \cup \dots \cup S_k) - (X_1 \cup X_2 \cup \dots \cup X_k)\}$ は空であるか? 空ならアルゴリズムは完了した。空でないなら次のステップへ。

(5) $\{S_1 \cup S_2 \cup \dots \cup S_k\} - \{X_1 \cup X_2 \cup \dots \cup X_k\}$ の中から X_{k+1} を決定する。ステップ (3) へ。
 このアルゴリズムに従う端点探索の様子は第 5 図に示してある。



第 5 図 アルゴリズムの実行される様子

3.2 ラベリング技法

(a) ラベルの説明

前章の 3.1 で述べたアルゴリズムを実行するために、四種類のラベル $\{0, \alpha, \beta, \lambda\}$ を用いる。これらのラベルについて説明する。

今、アルゴリズムのステップ (5) で S_k, X_k まで求められたと仮定する。また $\alpha(\mathbf{x}^*)$ と書いて、 X のうち α とラベル付けられた端点を表わすことにする。 $0(\mathbf{x}^*), \beta(\mathbf{x}^*), \lambda(\mathbf{x}^*)$ についても同様の意味とする。

ラベル $[0]$; $0(\mathbf{x}^*) = \{\mathbf{x}^* | \mathbf{x}^* \in \bigcup_{i=1}^k S_i, \text{ or } \mathbf{x}^* \in X_k\}$

ラベル $[\alpha]$; $\alpha(\mathbf{x}^*) = \{\mathbf{x}^* | \mathbf{x}^* \in \bigcup_{i=1}^k S_i - \bigcup_{i=1}^k X_i\}$

ラベル $[\beta]$; $\beta(\mathbf{x}^*) = \{\mathbf{x}^* | \mathbf{x}^* \in X_k\}$

ラベル $[\lambda]$; $\lambda(\mathbf{x}^*) = \{i | J_i \in J\}$

ラベル $[0]$ は、このラベルの付けられた端点に関する情報はこのステップでは必要のないことを示している。すなわち、(i) まだ探索されていない端点、又は (ii) すでに X_1, \dots, X_{k-1} のどのレベルかに属していて、且つその隣接端点を探し終った端点、に付けられる。一般のラベリング法では (ii) のように以前のラベルを消してしまうような事は起らないが、この問題の場合は目的関数の値が大きい方から順に探索して来ているのでこのような方法が許される。

ラベル $[\alpha]$ はこのステップで新しく隣接隣点として求められた端点のうち、まだ X_1, \dots, X_k のどのレベルにも属していない端点に付けられる。ラベル α の端点は次のレベル分けのとき新しいレベルの端点として候補にあがる。(ラベル α の端点のうち次のステップで新しく $(k+1)$ レベルの端点となる端点は α から β へとラベルを付け変えられる)

ラベル $[\beta]$ は新しくレベルに分類された端点に付けられ、このラベルの端点は次に隣接端点を求めるときの基となる。(隣接端点を求めた後、 β ラベルの端点は β から 0 へとラベルを付け変えられる)

基底解はそれを構成するベクトルが同じであるなら同じ解となる。例えば、基底解 \mathbf{x}_1^* と \mathbf{x}_2^* について、 $J_1 = J_2$ であるなら $\mathbf{x}_1^* = \mathbf{x}_2^*$ である。従って、ある基底添字集合 J_i に対して J_i の要

素のすべての組合せ $J_i^1, J_i^2, \dots, J_i^k$ に対して同一の解 x_i^* が対応している。よって、ある組合せである J_i^l についての解が得られたなら他の組合せ $J_j^i (j \neq l)$ については新たに解を求める必要がない。この目的のためにラベル λ を用いる。

(b) ラベリング・アルゴリズム

ラベル $\{0, \alpha, \beta, \lambda\}$ を用いたラベリング・アルゴリズムは次のように実行される。

(1) シンプレックス法により x_0^0 を求める。 x_0^0 を求める途中で得られた端点にはラベル α を、 x_0^0 にはラベル β を付ける。また、それぞれの基底添字集合 J_i にはラベル λ を付ける。

(2) ラベル β の端点の ALTERNATIVES を求め、ラベル β を付ける。これを繰り返し、 $J_i \in J$ である ALTERNATIVES がなくなったら (3) へ。(当然 ALTERNATIVES に対しての J_i にも λ を付ける)

次の (3) から (5) は繰り返しの手順となる。

(3) ラベル β の端点に対して、その隣接端点があればそれを求める。このとき、その隣接端点の基底添字集合 J_i にラベル λ が付いている (すなわち、 $J_i \in J$) ならばその基底解を求める必要はない。新しく求めた隣接端点にはラベル α を、その基底添字集合 J_i には λ を付ける。ラベル β の端点はその隣接端点のあるなしにかかわらずラベル 0 に付け変える。

(4) ラベル α の端点が存在するか? なければアルゴリズムは終了。あれば次へ。

(5) ラベル α の端点の中から目的関数の値が最大の端点集合を決定し、これらの端点について α から β へとラベルを付け変える。(3) へ。

3.3 計 算 例

例 1 を用いてアルゴリズムの実行される様子を詳しく説明し、ALTERNATIVES が存在する例として、例 2 をあげる。また用いているタブローはコンパクト・タブローで次の型をしている。(表 1)。

例 1

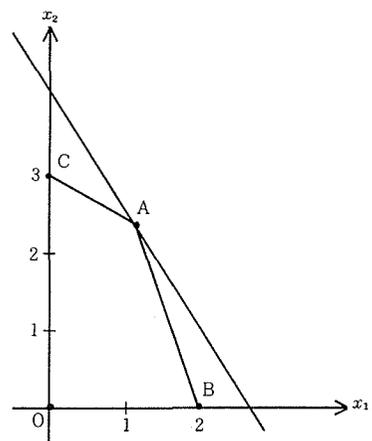
第 6 図を参照しながら説明する。

$$\begin{aligned} z &= 5x_1 + 3x_2 \\ \text{sub. to;} & 3x_1 + 5x_2 \leq 15 \\ & 5x_1 + 2x_2 \leq 10 \\ & x_1, x_2 \geq 0 \end{aligned}$$

(1)-1 シンプレックス法により $\max z$ を求める。このとき調べられる端点は $O \rightarrow B \rightarrow A$ で A が $\max z$ を与える端点である。 O, B, A に対するタブローは表 2-1 から表 2-3 に示す。 $0(x^*) = \{C\}$, $\alpha(x^*) = \{O, B\}$, $\beta(x^*) = \{A\}$ 。基

表 1 コンパクト・タブロー

非基底 ベクトル 基底ベクトル	f_1	f_2	...	f_n	x_B
b_1	y_{11}	y_{12}	...	y_{1n}	x_{B1}
\vdots	\vdots	\vdots	...	\vdots	\vdots
b_m	y_{m1}	y_{m2}	...	y_{mn}	x_{Bm}
$z_j - c_j$	$z_1 - c_1$	$z_2 - c_2$...	$z_n - c_n$	z



第 6 図 例 題 1

表 2-1 端 点 O

	a_1	a_2	x_B
a_3	3.00	5.00	15.00
a_4	5.00	2.00	10.00
$z_j - c_j$	-5.00	-3.00	0.00

表 2-2 端 点 B

	a_1	a_2	x_B
a_3	-0.60	3.80	9.00
a_4	0.20	0.40	2.00
$z_j - c_j$	1.00	-1.00	10.00

表 2-3 端 点 A

	a_1	a_2	x_B
a_2	-0.16	0.26	2.37
a_4	0.26	-0.11	1.05
$z_j - c_j$	0.84	0.26	12.37

表 2-4 端 点 C

	a_1	a_2	x_B
a_2	0.60	0.20	3.00
a_4	3.80	-0.40	4.00
$z_j - c_j$	-3.20	0.60	9.00

底ベクトルが $J_1 = \{3, 4\}$, $J_2 = \{3, 1\}$, $J_3 = \{2, 1\}$ からなる解にはラベル λ を付ける。 $J = \{J_1, J_2, J_3\}$ 。

(2)-1 A の ALTERNATIVES を求める。この例ではない。

(3)-1 A の隣接端点を探す。B と C であるが B はすでに求めてあるので捨てる。 $\alpha(x^*) = \{A\}$, $\alpha(x^*) = \{O, B, C\}$, $\beta(x^*) = \phi$, $J_4 = \{2, 4\}$, $J = \{J_1, J_2, J_3, J_4\}$ 。(表 2-4)

(4)-1 $\alpha(x^*) \neq \phi$ 。(5)へ。

(5)-1 $\alpha(x^*) = \{O, B, C\}$ の中から z の最大の端点を決める。B がそれである。 $\alpha(x^*) = \{A\}$, $\alpha(x^*) = \{O, C\}$, $\beta(x^*) = \{B\}$ 。(3)へ。

(3)-2 $\beta(x^*) = \{B\}$ であるから B と隣接している端点を探す。O であるが O はすでに求めてある。 $\alpha(x^*) = \{A, B\}$, $\alpha(x^*) = \{O, C\}$, $\beta(x^*) = \phi$ 。

(4)-2 $\alpha(x^*) \neq \phi$ 。(5)へ。

(5)-2 O と C との z 値を比べて大きい方の C を第 3 レベルの端点とする。 $\alpha(x^*) = \{A, B\}$, $\alpha(x^*) = \{O\}$, $\beta(x^*) = \{C\}$ 。(3)へ。

(3)-3 C の隣接端点 O を見つけるが、O はすでに求めてある。 $\alpha(x^*) = \{A, B, C\}$, $\alpha(x^*) = \{O\}$, $\beta(x^*) = \phi$ 。

(4)-3 $\alpha(x^*) = \{O\} \neq \phi$ 。(5)へ。

(5)-3 O が第 4 レベルの端点となる。 $\alpha(x^*) = \{A, B, C\}$, $\alpha(x^*) = \phi$, $\beta(x^*) = \{O\}$ 。(3)へ。

(3)-4 すべての端点のラベルは o となる。 $\alpha(x^*) = \{A, B, C, O\}$, $\alpha(x^*) = \phi$, $\beta(x^*) = \phi$, $J = \{J_1, J_2, J_3, J_4\}$ 。

(4)-4 $\alpha(x^*) = \phi$ 。stop。

例 2

第 7 図について説明する。

$$z = -x_1 - 4x_2$$

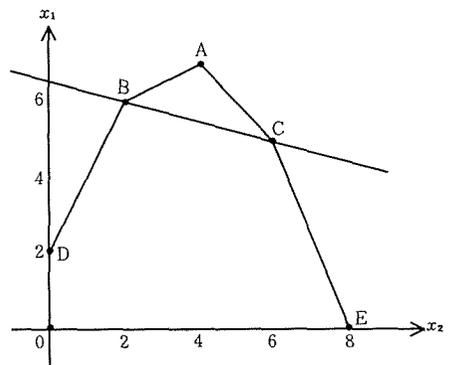
$$\text{sub. to; } -x_1 + 2x_2 \leq 2$$

$$x_1 + x_2 \leq 11$$

$$5x_1 + 2x_2 \leq 40$$

$$x_1, x_2 \geq 0$$

(1)-1 $O \rightarrow D \rightarrow B \rightarrow A$ 。 $\alpha(x^*) = \{C, E\}$, $\alpha(x^*) = \{O, D, B\}$, $\beta(x^*) = \{A\}$ 。 $J_1 = \{3, 4, 5, 6\}$, $J_2 = \{2, 4, 5, 6\}$, $J_3 = \{2, 1, 5, 6\}$, $J_4 = \{2, 1, 3, 6\}$ 。



第 7 図 例 題 2

表 3-1 端 点 C

	a_5	a_6	x_B
a_2	-0.33	1.67	5.00
a_1	0.33	-0.67	6.00
a_3	1.00	-3.00	9.00
a_4	1.00	-4.00	6.00
$z_j - c_j$	-1.00	6.00	26.00

表 3-2 端 点 B

	a_1	a_5	x_B
a_2	0.67	-0.33	6.00
a_1	0.33	-0.67	2.00
a_5	-1.00	1.00	3.00
a_6	-3.00	4.00	18.00
$z_j - c_j$	3.00	-2.00	26.00

(2)-1 ALTERNATIVES なし。次へ。

(3)-1 $\alpha(x^*) = \{A, C, E\}$, $\alpha(x^*) = \{O, D, B, C\}$, $\beta(x^*) = \phi$, $J_3 = \{2, 1, 3, 4\}$ 。

(4)-1 $\alpha(x^*) \neq \phi$ 。(5)へ。

(5)-1 $\alpha(x^*) = \{A, C, E\}$, $\alpha(x^*) = \{O, D\}$, $\beta(x^*) = \{B, C\}$ 。(表 3-1, 3-2)。(3)へ。

(3)-2 $\alpha(x^*) = \{A, B, C\}$, $\alpha(x^*) = \{O, D, E\}$, $\beta(x^*) = \phi$ 。

(4)-2 $\alpha(x^*) \neq \phi$ 。(5)へ。

(5)-2 $\alpha(x^*) = \{A, B, C\}$, $\alpha(x^*) = \{O\}$, $\beta(x^*) = \{D, E\}$ 。

(3)-3 $\alpha(x^*) = \{A, B, C, D, E\}$, $\alpha(x^*) = \{O\}$, $\beta(x^*) = \phi$ 。

(4)-3 $\alpha(x^*) \neq \phi$ 。(5)へ。

(5)-3 $\alpha(x^*) = \{A, B, C, D, E\}$, $\alpha(x^*) = \phi$, $\beta(x^*) = \{O\}$ 。

(3)-4 $\alpha(x^*) = \{A, B, C, D, E, O\}$, $\alpha(x^*) = \phi$, $\beta(x^*) = \phi$ 。

(4)-4 $\alpha(x^*) = \phi$ 。stop。

4. あとがき

本報告で述べたアルゴリズムは非線型計画問題を解くための部分アルゴリズムとしての利用が考えられ、筆者等は固定値付き問題への応用を図っている。この問題については改めて報告したい。

参 考 文 献

- 1) Hadley, G.: Linear Algebra (1961), Addison-Wesley.
- 2) Hadley, G.: Linear Programming (1962), Addison-Wesley.
- 3) Hadley, G.: Non Linear and Dynamic Programming (1964), Addison-Wesley.
- 4) 加瀬滋男, 森 健一: コンピューターと OR (1970), 日刊工業新聞社。
- 5) Steinberg, D. I.: *Nav. Res. Log. Quart.*, 17 (1970) pp. 217-235.
- 6) Kirby, M. J. L., etc.: *Management Sci.* 18 (1972), 9, p. 540-549.