幾何計画法における近似分解アルゴリズム*

吉 田 要**

Decomposition Algorithm in the Approximate Optimization Method of Geometric Programming*

Kaname YOSHIDA**

In this paper, we propose an algorithm to decompose a geometric programming problem into a set of subproblems. Each of the subproblems is constructed with some column vectors that are chosen from the $n \times m$ exponent matrix of the given problem to satisfy each of the following :

- (1) the convex hull of the selected $n+1$ vectors contains zero vector as an interior point ;
- (2) every column vector of the exponent matrix must be included in some set which consists of $n+1$ vectors that satisfy the condition (1).

The present paper represents a systematic choice algorithm of the column vectors. As the selection method is a pivoting method, the complexity of computation is very small.

At last some examples are dealt with to show the efficiency of this algorithm.

1. はじめに

最適化問題を効率よく解くためには、問題の構造をうまく利用した最適化法が考案されることが多い。そのようなものの一つとして幾何計画法 (Geometric Programming) がある¹⁻¹¹。幾何計画法は多項式形の制約関数と目的関数を持つ最適化問題で、非線形最適化問題の重要な手法の一つとなっている。この問題を、幾何不等式 (Geometric Inequality) を利用することによって線形制約をもつ最適化問題に変換することができ大変扱い易くなる。原問題 (Primal Problem) は最小化問題であるが変換後の問題は最大化問題となり、双対問題 (Dual Problem) と呼ばれる。両者の最適値は一致するので、扱い易さの点から通常は原問題に代えて、双対問題を解くことが多い。この双対問題を解く場合の計算量をはかるものとして、難度 (Degree of Difficulty) がある。たまたま難度が0の問題においては、双対問題の許容解はただ一つとなり、簡単に解がえられることに

なる。この場合でも、原問題の許容解は一意でないので、原問題で扱う場合は、非線形最適化法によって、解の探索を行なう必要がある。このように、幾何計画法は、非線形最適化問題における興味深い手法の一つである。しかしながら、難度の大きい問題に対しては、このような手法を用いても、解を求めるのは大変であり、種々の工夫がなされている。

このような問題に対してよく利用されているものとして、分割手法 (Decomposition Method) がある。適当な方法を用いて、原問題と等価ないくつかの低難度問題に分解し、低難度問題を解くことによって元の原問題の解を求めるものである。しかし、一般には等価な低難度問題に変換できる場合とそうでない場合があり、変換できない場合にはなんらかの工夫が必要で、一つの考え方は近似的に解くことによって原問題の有効な許容解あるいは近似解を得るというものである。

一方、ある種の物理的なシステムの最適化問題においては、構造はほとんど変化しないけれども問題に含まれる一部のパラメータが変化した問題を反復的に扱うことがある。このような場合は、有効な初期解が、前回までに扱った問題の解やそのほかの知識から、推測できることがある。また、可能なシステムが複数個あり、ある条件を満足するものであれば、最適なものでもなくてもよい

* 原稿受付 1989. 6. 9

** 立命館大学 経営学部 Faculty of Business Administration, Ritumeikan University ; Kita-ku, Kyoto 603, JAPAN

Key Words : geometric programming, decomposition method, optimization, approximation, algorithm.

というような場合、厳密解を求める代わりに有効な許容解が得られれば、それらを利用して、とりあえず有効なシステムを決定し、それに対して厳密に解ければよいという考え方が可能となる。

以上の観点から、著者は、与えられた問題をいくつかの0難度問題に近似分解して扱う手法を提案している¹⁰⁻¹²。そこでは、近似的に0難度問題に分解できる問題のクラスを決定し、一定の条件(難度-評価関数に含まれる項の総数+1が0以下)を満足していれば、確率1で0難度問題に近似分解できることを示した。その後、近似分解した解から、元の問題における近似解を構成する方法と、評価法について示した。ところで、0難度問題を構成するためには、原問題を構成する多項式に含まれる指数行列(Exponent Matrix)から、いくつかの0難度問題を構成するのに必要な指数ベクトル(Exponent Vector)の組を決定しなければならない¹⁰⁻¹²。すなわち、 n 次元問題を0難度問題に近似分解するためには、次のようなことが必要である。まず、 n 行 m 列($m \geq n+1$)の指数行列から適当な $n+1$ 個の列ベクトルを選ぶ。単に、 $n+1$ 個の列ベクトルの組を選ぶだけであるならば、選び方自体は、 ${}_m C_{n+1}$ 通りあることになる。しかし、そのうち必要なのは、選ばれた $n+1$ 個の列ベクトルによって構成される凸包が0ベクトルを含むようなベクトルの組である。そのような組のいくつかを選んだとき、原問題の指数行列の各列ベクトルは、選んだベクトルの組のいずれかに入っていることが必要である。多くの組を選べば、指数行列をカバーし易いことは確かであるが、組ごとに0難度問題をつくるため、指数行列をカバーするベクトルの組は少ない方がよい。このため、ベクトルを重複して採用することを極力避けることが望ましい。しかも、選んだベクトルの組の凸包が0ベクトルを含むことをチェックしなければならない。選んだベクトルの組について、すべて行なわなければならない、不要なベクトルを選ばないような効果的なアルゴリズムが必要である。これまでは、可能なベクトルの組が同じ組み合わせとならないようにしながら、とりあえず $n+1$ 個のベクトルの組を選び、この組に対して単体ベクトル集合であるかどうかをチェックしていた。このため、単体ベクトル集合を構成する有効なベクトルの組の選択ができず、最悪の場合は m 個から可能な $n+1$ 個のベクトルの組、すなわち、 ${}_m C_{n+1}$ 通りの組に対してチェックを行なう必要があった。

本論文では0難度問題のベクトルの組を決定する問題を多面体の分割問題として一般化して扱い、いくつかの有効な結果を得た。この結果を、0難度問題を構成するベクトルの組の探索アルゴリズムの構成に応用する。新

たに提案するアルゴリズムは、 $n+1$ 個のベクトルの組を選択する過程で不要なものと有効なベクトルとを振り分けることにより効率的に必要なベクトルの組を得ることができるシステムティックな方法である。最後に、本アルゴリズムが有効であることを計算例によって示す。

2. 多面体の単体分割

以下の準備のために、若干の記号と用語の定義を行っておく。

【記号】

ベクトル $a_i, i = 1, \dots, m$

$$a_i = (a_{i1}, a_{i2}, \dots, a_{in})^T$$

n 次元列ベクトル $a_1, a_2, \dots, a_m$ からなる行列 A を

$$A = [a_1, \dots, a_m] \quad m \geq n+1$$

と表わし、同時に A で

$$A = \{a_1, \dots, a_m\}$$

なるベクトル集合を表わすとする。また、 $n+1$ 個の列ベクトル $a_i \in A$ からなるベクトルの集合を $B \subset A$ と表わし、 B は A の部分行列でもある。

行列 B の列ベクトルの凸包を

$$\text{conv } B = \{b | b = B\lambda, B = [b_1, b_2, \dots, b_{n+1}], \\ \lambda = (\lambda_1, \lambda_2, \dots, \lambda_{n+1})^T, \sum_{i=1}^{n+1} \lambda_i = 1, \lambda_i \geq 0\}$$

とし、 A の凸包を $\text{conv } A$ とする。

【定義1】

$n+1$ 個の n 次元ベクトルの集合 B が n 単体をつくるとは、 B の凸包が内点を持つことである。すなわち、

$$\text{int}(\text{conv } B) = \{\text{conv } B \text{ の内点}\} \neq \phi$$

である。

【定義2】

$n \times m$ の行列 A が一般の位置にあるとは A の任意の $n+1$ 個のベクトルの集合が n 単体をつくることにいうとする。すなわち、任意の $n+1$ 個の列ベクトルからなる $B \subset A$ にたいして

$$\text{int}(\text{conv } B) \neq \phi$$

である。

【定義3】

$n \times m$ 行列 A の適当な $n+1$ 個の列ベクトルの組が n 単体を構成し、この凸包が0ベクトルを内点として含むとき単体ベクトル集合と呼ぶことにする。

【定義4】

単体ベクトル集合の組が与えられ、その和集合が行列 A のすべての列ベクトルを含むとき、与えられた単体ベクトル集合の組は A をカバーするといい、 A は単体

— 23 —

において, a_{r+1}' から a_m' までのどの列ベクトルを選ばよいであろうか. これに対しては, 次の結果が利用できる.

【定理】

n 行 m 列 ($m \geq n+1$) の行列 A と 0 ベクトルからなる $C = \{A, 0\}$ は一般の位置にあり, $0 \in \text{conv } A$ であるとする. このとき, 以下のことが言える.

[1] 指数行列 A の任意の列ベクトルはある単体ベクトル集合の要素となっている.

[2] もし, 第1行から第 r ($< n$) 行まで, 対角化が進んだとき, $a_1, \dots, a_r$ がある単体ベクトル集合に含まれるためには, $r+1$ 列から m 列までの列ベクトルについて, 次のことが言えなければならない.

(2-1) 1行から r 行までのすべての行において, 少なくとも一つは負要素を含まねばならない.

(2-2) $r+1$ 行から n 行では, すべての行で異符号の要素をもたねばならない.

[3] $a_1, \dots, a_r$ を含む単体ベクトル集合が存在するとする. いま, 第1行から第 r 行までのある行で, 負の要素をただ一つ含むような行があれば, その負要素を含む列は $a_1, \dots, a_r$ を含む単体ベクトル集合に含まれる.

また, 第 $r+1$ 行から第 n 行までで異符号のものがただ一つであれば, 0 でない異符号の要素を含む列ベクトルは $a_1, \dots, a_r$ を含む単体ベクトル集合に含まれる.

[4] もし, n 個のベクトルを対角化の後, ある列ベクトルの要素がすべて負であれば, その列ベクトルと対角化に利用した n 個の列ベクトル集合は単体ベクトル集合となる.

[5] このようにして得られる単体ベクトル集合の組は, 行列 A を単体分解するだけ得られる.

証明の前に, 上の [1]~[5] を簡単に説明しておく.

[1] は任意のベクトルに対して, 少なくとも一つはそのベクトルを含むような単体ベクトル集合があることをいっており, このことが単体分解する上で重要である.

[2] は順次単体ベクトル集合に含まれると思われるものを選んだときに, ある時点で, 選んできたベクトルを含む単体ベクトル集合は存在しないことが分かる条件である. そして, このことが起こるのは, [3] の条件がはずれる場合, すなわち, 1から r 行までで負の要素が複数個あり, かつ $r+1$ から n 行までで異符号が複数個ある場合である. この場合は単体ベクトル集合の要素として, [2] の必要条件を満足するものが複数あり, その中から選んだために, 不用なものを選ぶ可能性がある. その場合, 対角化の途中で, 次の単体ベクトル集合の要素が選ばない場合が出てきて, その時選んだベクトルが不都合であることが判明するわけである. [4] は最終的に, 単

体ベクトル集合が確定する条件である. また, [5] はこのようにしてつくられた単体ベクトル集合は A をカバーするだけ, すなわち単体分解できるだけ選べることを意味している.

【定理の証明】

[1] 補題の結果と, A と 0 ベクトルが一般の位置にあることから, A のいかなる n 個の列ベクトルに対してもその凸包には含まれないことから言える.

[2] (1) 式において, 第1行から第 r 行までで, 第 $r+1$ から第 m 列までの部分行列を考える. この中のある行 i ($1 \leq i \leq r$) の要素がすべて, 正であったとすると, $a_1', \dots, a_r'$ (すなわち $a_1, \dots, a_r$) を含むある単体ベクトル集合, 仮に $a_1', \dots, a_{n+1}'$ (すなわち $a_1, \dots, a_{n+1}$) とする. 単体ベクトル集合の凸包は 0 ベクトルを含むから, 0 ベクトルを構成する結合係数 f に 0 要素をつけ加えてできる係数ベクトル F を

$$F = (f, 0, \dots, 0)^T$$

行列 (1) に乗ずれば, i 行は

$$f_i + \sum_{j=r+1}^{n+1} a_{ji} f_j > 0$$

となり, $a_1, \dots, a_{n+1}$ が単体ベクトル集合であることに反することになる. [2] の後半部分についても同様である.

[3] については, [2] の証明より明らかである. [4] は明らか. [5] は [1] から, すべてのベクトルに対して, 単体ベクトル集合が構成できるのであるから, 行列 A をカバーできるだけ得ることができる.

【証明終わり】

アルゴリズムで使用する添字集合として以下のものを定義しておく.

S : 確定している単体ベクトル集合の要素となっているベクトルの添字集合

Q : 構成しつつある単体ベクトル集合の要素として選択されているベクトルの添字集合

R : 単体ベクトル集合を構成しつつあるとき, 不都合であると判定されたベクトルの添字集合

以上の添字集合を用いて, アルゴリズムの概略を説明しておく.

まず, 定理の [1] から, 任意にベクトル (仮に a_1) を選び, その添字を Q の要素とする ($Q = \{1\}$). 選んだベクトル a_1 で対角化を行ない, 定理の [2], [3] の条件を満たす (単体ベクトル集合の要素となる可能性のある) ベクトル (a_i) を選ぶ. 新たに選んだベクトル a_i で対角化を続行し, 再び定理の [2], [3] の条件を満たすベクトル (a_i) が存在すれば添字 i を Q に追加する ($Q = \{1, i\}$).

このプロセスを繰り返し、 Q に含まれる要素の数が n となったとき、 Q に含まれている添字に対応するベクトルの組が単体ベクトル集合を構成するかどうか判定される。もし、定理の[4]の条件によって構成することが判明すれば、 Q とあと一つの要素(k)が選択できるのでそれを合わせて、 S に加える($S = \{Q, k\}$)。もし、途中で対角化プロセスが続行できない場合は直前の対角化ベクトルの添字を R の要素として追加していく。次の対角化要素は S, R, Q に含まれるものは除いて選択する。このプロセスを繰り返すことによって、重複せずにしかも有効な組を効率的に選択することができる。そして、定理の[5]の結果から、 S は L のすべての要素を含むまで実行でき、含むことが判明した時点で A は単体分解されたことになり、計算をstopさせることになる。

以上の事をアルゴリズムで示すと、次のようになる。

【単体ベクトル集合の決定アルゴリズム】

- step 1: 行列 $A = \{a_1, a_2, \dots, a_m\}$ の列ベクトルの添字集合 $L = \{1, 2, \dots, m\}$, $S = \phi$ としてセットし、step 2 へ
- step 2: $L - S = \phi$ なら stop. さもないければ、 $Q = \phi, R = \phi, i = 1$ とし step 3 へ、
- step 3: $j \in L - S - R$ として A の i, j 列を入れ替えて j 列で角化して、 Q の要素の数が n ならば step 4 へ、 n でなければ step 6 へ
- step 4: 定理の[4]の条件を満足する k があれば k と j を Q に加えて、 S に Q を追加して step 2 へ、満足するものがなければ step 5 へ
- step 5: R に j を加えて、step 3 へ
- step 6: 定理の[3]の条件を満足するものを調べ、あれば Q に j を加えて、 $i = i + 1$ として step 3 へ、定理の[3]の条件を満足するものがなければ、定理の[2]の条件を満足するものを調べる。定理の[2]の条件を満足するものがあれば $i = i + 1$ として step 3 へ、定理の[2]の条件もなければ j を R に加えて step 3 へ

4. 幾何計画問題への応用

上で得られたアルゴリズムを幾何計画法の近似最適化問題に応用する。与えられた問題を0難度問題で近似する手法については、参考文献10)~12)において報告し、有効性を示している。しかしながら、0難度問題を構成するためには、指数行列の m 個の列ベクトルから0難度問題を構成できる $n+1$ 個の列ベクトルの組を見つけ、指数行列全体をカバーするように取る必要がある。2. で扱った行列 A を指数行列とし、単体ベクトル集合を0難度問題の指数行列として扱うことができる。

4.1 幾何計画法

幾何計画法は次のような問題を扱う。

【主問題】(Primal Problem)

$$\begin{aligned} \min_{x \geq 0} G_0(x) &= \sum_{i \in L(0)} g_{0i}(x) \\ \text{sub. to } G_k(x) &= \sum_{i \in L(k)} g_{ki}(x) \leq 1 \\ k &\in P = \{1, 2, \dots, p\} \end{aligned}$$

ただし $x = (x_1, x_2, \dots, x_n)^T \in R^n$

C_{ki} : 係数で正の数

a_{kij} : x_j の指数である。

$$g_{ki}(x) = C_{ki} \prod_{j=1}^n x_j^{a_{kij}} = C_{ki} x^{a_{ki}}$$

$a_{ki} = (a_{ki1}, \dots, a_{kin})^T \in R^n$: 指数ベクトル

$q = \sum_{k=0}^p l(k)$: 主問題に含まれる項の総数

$l(k)$: 制約式 $G_k(x)$ に含まれる項の総数

$L(k) = \{1, 2, \dots, l(k)\}$: 添字集合

【双対問題】(Dual Problem)

$$\begin{aligned} \max_{\delta \geq 0} v(\delta) &= \prod_{k=0}^p \left\{ \prod_{i \in L(k)} \left[\frac{C_{ki} \lambda_k}{\delta_{ki}} \right]^{\delta_{ki}} \right\} \\ \text{sub. to } A\delta &= 0 && \text{直交条件} \\ \lambda_0 &= \sum_{i \in L(0)} \delta_{0i} = 1 && \text{正規化条件} \end{aligned}$$

ただし、 $\lambda_k(\delta) = \sum_{i \in L(k)} \delta_{ki}, k \in P$

$$\delta = (\delta_{01}, \delta_{02}, \dots, \delta_{p(p)})^T$$

$A = [a_{01}, a_{02}, \dots, a_{p(p)}] \in R^{n \times q}$: 指数行列

この指数行列に対して、単体分割を行ない、0難度問題を構成する。主問題が意味のある問題であれば、指数行列 A の $\text{conv } A$ が 0 を含むことは成り立つ。また、 A が一般の位置にあることは確率1で成り立つので、その様に仮定しても一般性は失われない。さらに、 $q - n - l(0) \leq 0$ なる条件を満たしていれば、0難度問題に分解できるので、2. の補題が適用でき本アルゴリズムが適用できる。一方、単体ベクトル集合を求める際に得られる係数ベクトル、すなわち定理の4で現われるすべての要素が負となる列ベクトルに、 $n+1$ 番目の要素として -1 をつけてできる $n+1$ 次元ベクトルは、原問題の評価関数に含まれる項に対応する係数を正規化する事によって、対応する0難度問題の双対最適解となる。

4.2 計算例

アルゴリズムを具体例に適用する。

【例】船の長期チャータに関する問題で、荷役能力の既知の二つと港の間で、ある物資を一定量/年輸送する場合の最小経費問題について考える。

変数は船の自重容積 D , 速度 V , 利用率 U , 船の数 N の4変数である。評価関数の第1, 2, 3項はそれぞれ、

建造費、機関の設備費、燃料費であり、制約式の第1, 2式はそれぞれ船の積荷に対する処理能力と航海時間および停泊時間に対する制約である。評価関数および制約関数に含まれる項の総数は7で、以下のようにになっている。

$$\min_{x \geq 0} G_0(x) = 200ND + (ND^{2/3}V^3)/2 + (4ND^{2/3}V^3U)/5$$

sub. to

$$G_1(x) = 2.75 \times 10^6 (NDVU) - 1 + 2V^2D^{-1/3} \times 10^{-3} \leq 1$$

$$G_2(x) = 0.67 \times 10^{-6} DVU + U \leq 1$$

この問題の難度は2である。

指数行列 A は次のようになる。

1	2	3	4	5	6	7	列/変数
1	1	1	-1	0	0	0	N
1	0.67	0.67	-1	-0.33	1	0	D
0	3	3	-1	2	1	0	V
0	0	1	-1	0	1	1	U

この問題で $n+1$ 個の列の選び方は ${}_7C_5$ で21通りあり、それらすべてが単体ベクトル集合となるとは限らない。しかも、指数行列をカバーするにとらねばならない。本論で提案する方法によれば大変簡単に処理することができる。

ところで、この例題の指数行列 A は、列2, 3, 7が1次従属になり、 A と0ベクトルが一般の位置にあるという定理の条件を満足しない。この条件は、任意に選んだ列ベクトルに対して、そのベクトルを含む単体ベクトル集合が存在する十分条件になっているため、この条件を満足しない場合は、最初のベクトルの選択が誤りである場合がでてくる。しかし、実際には A をカバーする組について解が存在すればよく、条件はもっと緩くて良いことになる。しかし、あらかじめ単体ベクトル集合の候補は決められないことと、一般の位置にあるという仮定は、多くの問題で満足される条件ということで仮定として採用している。この例題のように条件を満足しない場合でも、1次従属になるようなベクトルの組はアルゴリズムによって避けられるので、可能な組の探索ができることになる。よって、条件が満足していなくてもアルゴリズムに従えば、単体ベクトル集合になるような組を選んでいくことが可能であり、カバーするまでは可能な組をすべて探索することができる。アルゴリズムに従って、可能な組をすべて探索しても指数行列 A をカバーできない場合は A が分解できない場合であることは明らかである。

【計算例】

最初の行は列番号を書いている。最後の列には、アル

ゴリズムによって選択される列が候補列として示されている。*は候補列が複数個ある場合である。成功は単体ベクトル集合が確定した場合に示している。失敗は、 n 個の列ベクトルを対角化した結果、単体ベクトル集合とならないことが判明したときを示している。

まず、step 1 より、 $S = \phi$, $L = \{1, 2, 3, 4, 5, 6, 7\}$ とし step 2 へ、 $Q = \phi$, $R = \phi$ とし step 3 へ

1	2	3	4	5	6	7	
定理の [1] によって、第1列対角化							候補列
1	1	1	-1	0	0	0	(4)
0	-0.33	-0.33	0	-0.33	1	0	(6)
0	3	3	-1	2	1	0	(4)
0	0	1	-1	0	1	1	(4)

step 6 で4, 6列が該当するものである。4列と6列のいずれを選択してもよい。このように複数の候補列が存在する場合は、その内の一つを選ばばよい。ここでは、 $Q = \{1\}$ とし、step 3 へ移り、6列を選択する。step 6 へ

1	6	3	4	5	2	7	
定理の [3] によって、第6列対角化							
1	0	1	-1	0	1	0	(4)
0	1	-0.33	0	-0.33	-0.33	0	(*)
0	0	3.33	-1	2.33	3.33	0	(4)
0	0	1.33	-1	0.33	0.33	1	(4)

step 6 で4列が該当するので、 $Q = \{1, 6\}$ とし、step 3 へ、4列を選択し step 6 へ

1	6	4	3	5	2	7	
定理の [3] によって、第4列対角化							
1	0	0	-2.33	-2.33	-2.33	0	(*)
0	1	0	-3.30	-3.30	-3.30	0	(*)
0	0	1	-3.33	-2.33	-3.33	0	(*)
0	0	0	-2.0	-2.0	-3.0	1	(7)

step 6 で、7列が該当する。 $Q = \{1, 6, 4\}$ とし、step 3 へ移り、7列を選択。step 6 へ移る。

1	6	4	7	3	5	2	成功
定理の [3] によって、第7列対角化							
1	0	0	0	-2.33	-2.33	-2.33	
0	1	0	0	-3.3	-3.3	-3.3	
0	0	1	0	-2.33	-3.33	-3.33	
0	0	0	1	-2	-3	-2	

step 6 では3, 5, 2列が該当する。 $Q = \{1, 6, 4, 7\}$ とし、 Q の要素数は4なので step 4 へ移り、 $Q = \{1, 6, 4, 7, 3\}$ とする。同時に、 $Q = \{1, 6, 4, 7, 5\}$, $Q = \{1, 6, 4, 7, 2\}$ 。

$S = \{1, 6, 4, 7, 3, 5, 2\}$ で, step 2 へ移り, $L - S = \emptyset$ となりアルゴリズムは停止する.

このことから, 単体ベクトル集合として

$(1, 6, 4, 7, 5), (1, 6, 4, 7, 2), (1, 6, 4, 7, 3)$

が得られる. この三つの組み合わせで指数行列がカバーできるので, この例題は 0 難度分解できることがわかる. しかし, 列ベクトルの数は 7 で単体ベクトル集合は 5 個のベクトルの組からなっている. それゆえ, 可能性として二組の単体ベクトル集合で 0 難度解集合を構成できる可能性がある. いま, $(1, 6, 4, 7, 5)$ を上の結果から選んだとすれば, 2, 3 を含むものがとればよい. しかし, 列 2, 3 を同時に含むような単体ベクトル集合は存在しないことが分かる.

次の計算は 2, 4, 1 と選んで, 4 番目の列ベクトルとして 3 を選んだ結果を示している. 4 番目の選択は, 補題の条件にあわないが強引に選んだ結果を示している.

2 4 1 3 5 6 7 失敗
第 3 列対角化, 定理の [4] から単体ベクトル集合でないことが分かり 2, 4, 1, 3 を含むものはない

1	0	0	0	0	6.06	-1	(7)
0	1	0	0	1	-19.09	0	(6)
0	0	1	0	0	-7.06	0	(6)
0	0	0	1	1	-9.09	1	(6)

これは, $(2, 4, 1, 3, ?)$ の組み合わせでは単体ベクトル集合が存在しないことが分かる. 実際, このような不
用なケースはアルゴリズムに従えば避けることができる.
以下は, 第 2 列からアルゴリズムを適用した場合の結果
を示している. 第 2 列から初めて, アルゴリズムに従え
ば,

2	1	3	4	5	6	7
定理の [1] により, 第 2 列対角化						
1	1	1	-1	0	0	0 (4)
0	0.33	0	-0.33	-0.33	1	0 (*)
0	-3	0	2	2	1	0 (1)
0	0	1	-1	0	1	1 (4)
2	4	3	1	5	6	7

定理の [3] により, 第 4 列対角化

1	0	1	0	1	-3	0 (6)
0	1	0	-1	1	-3	0 (*)
0	0	0	-1	0	7	0 (1, 6)
0	0	1	-1	1	-2	1 (*)
2	4	6	1	5	3	7

定理の [3] により, 第 6 列対角化

1	0	0	-0.43	1	1	0 (1)
0	1	0	-1.43	1	0	0 (1)
0	0	1	-0.43	0	0	0 (1)
0	0	0	-1.28	1	1	1 (1)
2	4	6	1	5	3	7 成功

定理の [3] により, 第 1 列対角化

1	0	0	0	0.66	0.66	-0.33 (7)
0	1	0	0	-1.1	-1.11	-1.11 (7)
0	0	1	0	-1.1	-1.1	-1.1 (7)
0	0	0	1	-0.77	-0.77	-0.77 (7)

となって $(2, 4, 6, 1, 7)$ が選択され, 選択する過程で $(2, 4, 1, 3)$ を含むものを選ぶ可能性はなかったことが分かる.

5. おわりに

本論では, 0 難度問題に近似的に分解する際の分解法に関するシステムチックなアルゴリズムを示した. m 個の列ベクトルから $n+1$ 個の列ベクトルを選び, しかもそれらが単体ベクトル集合となるように選ぶことは, 選択法を誤ると大変効率の悪いものになるが本論での方法では, 効率よく選ぶことができ, 実例によって検証を行なった. 扱った例はいくつかあるが, いずれも効率よく選択することができた.

参 考 文 献

- 1) R. J. Duffin, E. L. Peterson & C. Zener : Geometric Programming Theory and Applications, John Wiley & Sons Inc., New York (1967)
- 2) C. Zener : Engineering Design by Geometric Programming, John Wiley & Sons Inc., New York (1971)
- 3) J. S. Folker : Ship Operation and Design, in Optimization and Design, Edited by D. J. Wilde, M. Avriel and M. J. Rijckaert, Prentice-Hall Inc. (1973)
- 4) D. J. Wilde & C. S. Beightler : Foundations of Optimization, Prentice-Hall Inc. (1967)
- 5) M. Heymann & M. Avriel : On a Decomposition for a Special Class of Geometric Programming Problems ; J. Opt. Theory and Application, Vol. 3, No. 6, pp. 389~409 (1969)
- 6) U. Passy & D. J. Wilde : Generalized Polynomial Optimization ; SIAM J. Appl. Math., No. 15, pp. 1344~1356 (1967)
- 7) 吉田, 平井 : 正多項式形幾何計画法の分割 ; システムと制御, 19 巻, 7 号, pp. 391~401 (1975)
- 8) 吉田, 児玉, 馬場 : 大規模幾何計画法問題の分割法 ; システムと制御, 20 巻, 12 号, pp. 689~697 (1976)
- 9) H. J. Greenberg & W. P. Pierskalla : Surrogate Mathematical Programming ; Operations Res., No. 18, pp. 924~939 (1970)
- 10) 吉田 : 幾何計画法の近似最適化法 ; 計測自動制御学会論文集, 19 巻, 8 号, pp. 624~649 (1983)
- 11) 吉田 : 幾何計画法問題の近似最適解の構成 ; 計測自動制御学会論文集, 20 巻, 2 号, pp. 130~136 (1984)

-
- 12) 吉田：幾何計画問題の近似最適化法における分解可能性,
計測自動制御学会論文集, 20巻, 7号, pp. 614~620 (1984)
-