クァドラティック・プログラミングの計算方法：タイル-バン・デ・パンの方法

クァドラティック・プログラミングの計算方法

――タイルーバン・デ・パンの方法――

今　川　　正

I　はじめに

経済学においてはクァドラティック・プログラミングはリニア・プログラミング以上の伸縮性，現実性を提供するために関心をもたれている．リニア・プログラミングにおいては，それぞれの活動の単位あたり費用（あるいは利潤）が固定されていると仮定されている．実際にはこの要因は固定されていないことが多い．

たとえば消費者が，購入数量が非負である，支出額合計が所得を越えないという制約条件のもとに最大化すべき効用関数は，数量とともに減少してゆく係数をもっている（限界効用逓減）．

また独占的生産者の場合もよく知られている．この企業のとりうる活動が n 種類あり，利用可能の資源が m 種類あるとする．活動 j の1単位によって消費される資源 i の数量を a_{ij} とする．そのとき S_i が利用可能の資源の全数量であるとし，活動 j の用いられる水準を x_j とすると，線型計画についてとまったく同じ制約条件

$$\sum_{j=1}^{n} a_{ij}x_j \leqq S_i \qquad i=1, \cdots, m$$
$$x_j \geqq 0 \qquad j=1, \cdots, n$$

をえる．この生産者の最大化すべき全収入は需要数量（したがって生産数量）x_j と価格 p_j の積であらわされる．そしてこの価格自身が数量の減少関数である（x に関する逆関数表示の需要関数）．このときにはうえの条件のもとに

$$\sum p_j(x_1, \cdots, x_n)x_j$$

を最大化するというノンリニア・プログラミングの問題をえる．ここに p_j が x_i に関して線型の関数であるとき，クァドラティック・プログラミングをえる．

クァドラティック・プログラミングの解をえるための計算方法にはいろいろのものがある．ラグランジュ乗数法のキューン，タッカーによる一般化にもとづくウォルフの方法，ダンチッヒの線型計画の展開によく似ているビールの方法，バランキン，ドーフマンのもの，タイルーバン・デ・パンのものなどがある．

ここではタイルとバン・デ・パンによって展開された方法について述べる．第Ⅱ節においてそれの概要を直観的に説明したあとで，第Ⅲ節においてその主要命題を代数的に展開する．第Ⅳ節で数値例を用いて計算のステップを詳細に述べ，第Ⅴ節をそのためのコンピューター・プログラムにあてる．他の方法に比較しての特徴などについては稿を改めて述べよう．

ここで考察するクァドラティック・プログラミングはつぎの形のものである．

$$\text{最大化}\qquad \phi(x)=a'x-\frac{1}{2}x'Bx$$

ここに B は $n\times n$ の正値定符号の対称行列である．この制約条件は

$$\text{条　件}\qquad C'x\leqq d$$

である．ここに C' は $m\times n$ の行列である．非負性条件のあるときには，それをこの条件の中に含めておく．(1)

（1）　等号の制約条件は除く．これを含むように拡張することは容易にできる　ブート〔3〕p. 109 をみよ．

クァドラティック・プログラミング問題を解く普通の方法は，制約条件を満足している任意の点x（これは必ずしも目的関数最大化の点ではない）よりはじめて，それを他のxで，――そのいずれも制約条件を満足している――目的関数の値を増大させるものとおきかえる．そしてもとめている解がえられるまでこれをつづける．これと，いわば逆の側より接近してゆく方法がある．そこにおいては最初はすべての制約条件を無視して目的関数を最大化する点xをもとめる．このようにしてえられたxは普通制約条件を満足していない．つぎに1つの制約条件を等式とおいて目的関数を最大化する点をもとめ，順に等式の形で用いる制約条件の個数を増加してゆく．ここで述べるのはこの後者の接近法についてである．

タイルとバン・デ・パンは制約条件が不等式の形であることが困難をひきおこしていると考えている．事実，ある目的関数を等式の制約条件のもとに最大化することはたいして難かしいことではない（ラグランジュ乗数法で容易に解くことができる）．うえで述べた不等式制約条件のもとに目的関数を最大化する解$\hat{x}$がもとまったとすると，制約条件の1部（あるいは全部）が等式の形で成立している．このグループを集合Sとよぶ．それで解をもとめるために彼らのサジエストする方法はm個の制約条件より，このような部分集合Sをもとめることと等価である．すなわち部分集合Sに属する制約条件をバインディングのものとして（すなわち等式の形で用いて）目的関数を最大化するときえられるx^Sが解ベクトルである．(2) $x^S=\hat{x}$．一般にx^SはSに属す

(2)　問題　最大化　$\phi(x)=a'x-\frac{1}{2}x'Bx$，条件　$c'x\leqq d$
の最適解を$\hat{x}$とする．この$\hat{x}$において制約条件は等式でみたされているものと，そうでないものとに分かれる．これを

$$c_S'\hat{x}=d_S$$
$$c_T'\hat{x}<d_T$$

と記す．この解$\hat{x}$は
問題　最大化　$\phi(x)=a'x-\frac{1}{2}x'Bx$，条件　$c_S'x=d_S$
の最適解　x^S　と同じである．

るすべての制約条件をバインディングとして目的関数を最大化するベクトルと定義する．ここにSはm個の制約条件の任意の部分集合である．Sに属する制約条件が等式の形で書かれるとき，斉合的であれば，このようなベクトルが存在する．ここではSにおいては x^S が存在しないことは起らないものと仮定する．

II　この方法の概要

タイル－バン・デ・パンの方法を代数的に展開するまえに，それを直観的に説明しておこう．彼らは空集合 ϕ およびベクトル x^ϕ の考察よりはじめる．ここに x^ϕ はどの制約条件も考慮に入れないで $\phi(x)$ を最大化するベクトルである．微分を用いて

$$x^\phi = B^{-1}a$$

を容易にもとめることができる．もしこの x^ϕ がすべての制約条件を満たしておれば x^ϕ がもとめている解である．制約条件付き最大が無制約条件の最大を超えることはありえないからである．しかしわれわれの関心をいだいている場合には x^ϕ が制約条件を侵害していることが多い．

図1を用いて説明しよう．これは $n=2$ の場合で，制約条件は非負性のもののみとする．$\xi_1 \geqq 0, \xi_2 \geqq 0$. 実行可能領域は第1象現である．斜線をつけてこれを示す．図においては目的関数は x^ϕ において無条件最大値に達している．

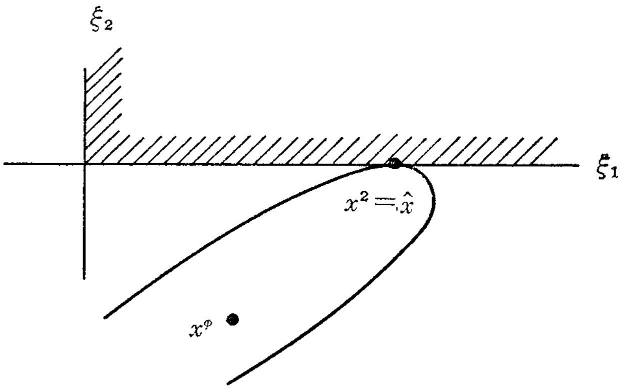

図　1

九七 x^ϕ のまわりの1つの同心楕円上で $\phi(x)$ の値は一定である．そしてその値は外にゆくにつれて減少する．Bが正値定符号のときにはこうなる．われわれは実行

可能領域にとどまったまま，もっとも内側の楕円に達しようとしている．図1の例においては楕円が横軸に接している点 x^2 においてこれに達する．x^ϕ は条件 $\xi_2 \geqq 0$ を侵害している．それで $\phi(x^\phi)$ より低い $\phi(x^2)$ の値で満足しなければならない．代数学的には x^2 は $\xi_2=0$ の条件のもとに，すなわち制約条件2を等式の形で課したうえで $\phi(x)$ を最大化することによってえられるベクトルである．このベクトルによって制約条件2は過不足なく，すなわち等式の形で満たされている．ここで考察しているものは最適ベクトルであるから，図1の観測をつぎのようにまとめることができる．無制約最大化ベクトル x^ϕ は制約条件の1つを侵害している．そして制約条件付き最大化のもの $\hat{x}$ は同じ制約条件を等式の形で満たしている．

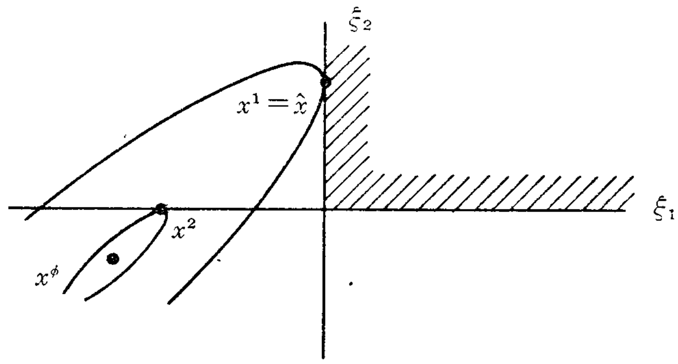

図　2

つぎに図2をみよう．ここでは x^ϕ が制約条件1，2を両方侵害している．まえの図1はわれわれのとるべき接近法が制約条件を順に等式の形で課すことをサジェストしている．$\xi_2=0$ のもとに最適化するとベクトル x^2 をえる．これは制約条件 $\xi_1 \geqq 0$ を侵害している．$\xi_1=0$ の条件のもとにえられるものを x^1 とする．これはどの制約条件も侵害していない．これは明らかに最適ベクトルである．この結果はうえで述べたルールをつぎのように拡張すべきことをサジェストしている．無制約条件最大化のベクトル x^ϕ が1つ以上の制約条件を侵害するとき，それのうち少なくとも1つが解においてバインディングである．しかし x^ϕ によって侵害されている制約条件のすべてが解においてバ

インディングとは限らない．

図3においては x^ϕ が制約条件2を侵害している．解 $x^{12}=\hat{x}$ においては制約条件1，2が両方バインディングである．それで x^ϕ によって侵害している制約条件の少なくとも1つが解においてバインディングであるといえる．しかし x^ϕ によって侵害されていない制約条件が解においてバインディングであることが可能である．

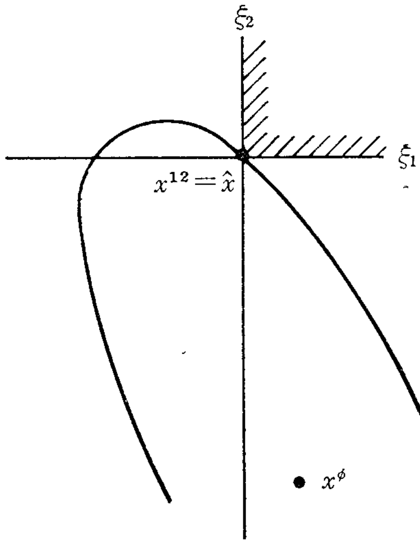

図　3

これまでにみてきたことはつぎのようにまとめられる．すなわち x^ϕ がある制約条件を侵害していると，この少なくとも1つが最適ベクトル $\hat{x}$ においてバインディングである[(3)]．

タイルとバン・デ・パンはうえに述べたルールにしたがって，まず x^ϕ によって侵害されている制約条件のインデックス i それぞれよりなる1要因集合 $\{i\}$ について考える．1個の制約条件を等式とおいてえられる最大化ベクトルのうちに実行可能のものがあればそれが解であろう．実行可能のものがなければ，つぎに2要因集合 $\{i, j\}$ について考える．ここに最初の要素は i に x^ϕ よって侵害されているもののインデックスであり，第2の要素 j は x^i によって侵害されている制約条件のインデックスである．このようにつづけてゆく．この接近法の基本的考えは，x^ϕ の侵害している制約条件の少なくとも1つが解においてバインディングであるという観測の一般化である．

この方法によって集合 S，そのもとでの x^S が実行可能であるものがもとまったとする．この x^S がわれわれの問題の解であるかという疑問が残る．それが解であるとは限らない．図4の例を用いて説明しよう．ここでは2つの非負性制約条件1，2と2つの付加的制約条件

（3）　非負性以外の制約条件のある場合にもこのルールは成立する．

3，4のある場合をえがいておいた．x^ϕ が1，3，4を侵害していることがわかる．それで x^1, x^3, x^4 について考える．x^1 は等式とおいた制約条件1のもとに目的関数を最化大するベクトルである．x^3, x^4 も同じような最大化ベクトルである．x^1 は制約条件3，4を侵害している．それでつぎのラウンドにおいて x^{13}, x^{14} について考える．また x^3 は1，4を侵害している．それでつぎに x^{31}, x^{34} について考える．そして x^4 は1を侵害している．それで x^{41} について考える(4)．ここで実行可能のものは，x^{14}, x^{34} である．しかし x^{14}, x^{34} が解であるか否かをチェックする仕事が残っている．

タイルとバン・デ・パンは実行可能ベクトル x^S が最適であるための必要かつ十分な条件として，すべての x^{S-t} が（$S-t$ は集合 S より要素 t を除いたものをあらわす）が制約条件を侵害することであると述べている．図4の場合には x^{34} は解でない．x^{34-4} は制約条件4を侵害するが，x^{34-3} は制約条件4を侵害しないからである．他方，x^{14} は解である．x^{14-1} は制約条件1を侵害し x^{14-4} は制約条件4を侵害するからである．

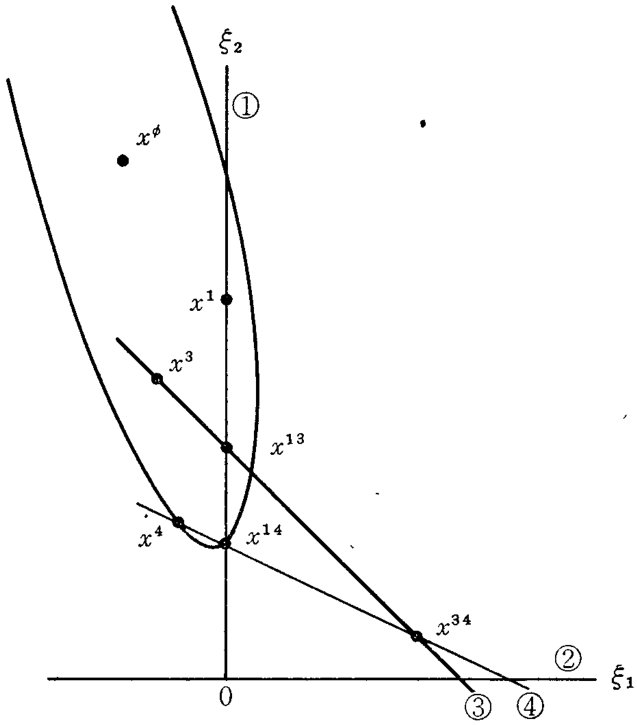

図　4

これがタイルとバン・デ・パンの方法の基本的考え方である．ここで制約条件を侵害しているか否かが，2つの段階において決定的規準として用い

(4)　制約条件1，4の集合をバインデングとするものと4，1の集合をバインデングするものとは同じである．

られている.[5]

簡単な例を用いてこの方法を説明しよう.

目的　最大化　$\phi(x)=10\xi_1+25\xi_2-10\xi_1{}^2-4\xi_1-\xi_2$

条件　1　$-\xi_1 \leqq 0$

　　　2　$-\xi_2 \leqq 0$

　　　3　$\xi_1+\xi_2 \leqq 9$

　　　4　$\xi_1+2\xi_2 \leqq 10$

この接近法においては最初にすべての制約条件を無視して問題を解く.

最大化　$\phi(x)=10\xi_1+25\xi_2-10\xi_2{}^2-4\xi_1\xi_2-\xi_2{}^2$

このためには微分してえるつぎの連立方程式を解く.

$$\frac{\partial\phi}{\partial\xi_1}=10-20\xi_1-4\xi_2=0$$

$$\frac{\partial\phi}{\partial\xi_2}=25-4\xi_1-2\xi_2=0$$

これを解くと解 $x^\phi=(-3.3, 19.1)$ 目的関数の値 $\phi=222.9$ をえる. この点 x^ϕ を図4に記しておいた. そこにみられるようにこの点は制約条件1, 3, 4を侵害している.

この結果にもとづいてつぎの3種の問題をつくる（以下いずれも $\phi(x)$ を最大化する問題であることはいうまでもない).

条件	$\xi_1=0$	問題1
条件	$\xi_1+\xi_2=9$	問題3
条件	$\xi_1+2\xi_2=10$	問題4

問題1を解くと $x^1=(0, 12.5), \phi=156.2$ をえる. これは制約条件3, 4を侵害している. 問題3を解くと $x^3=(-2.3, 11.3)$, $\phi=182.8$ をえる. これは制約条件1, 4を侵害している. 問題4を解いてえる解

（5）　タイルとバン・デ・パンはこれを〔6〕において証明している. そしてブートはこれと等価の命題をラグランジュ乗数の符号で述べている. そしてこの両者が等価であることはつぎの節で述べておく.

$x^4=(-1.0, 5.5)$，$\phi=109.2$ は制約条件 1 を侵害する．この結果を図 4 にえがいておいた．

つぎに 2 要因制約条件集合のもとに問題を解かねばならない．うえの問題 1 よりはつぎの 2 つの $\phi(x)$ 最大化の問題をえる．

条件　$\xi_1=0,\ \xi_1+\xi_2=9$　問題 1，3

条件　$\xi_1=0,\ \xi_1+2\xi_2=10$　問題 1，4

問題 3 よりはつぎの 2 つの問題をえる．

条件　$\xi_1+\xi_2=9,\ \xi_1=0$　問題 3，1

条件　$\xi_1+\xi_2=9,\ \xi_1+2\xi_2=10$　問題 3，4

問題 4 よりつぎの問題をえる．

条件　$\xi_1+2\xi_2=10,\ \xi_1=0$　問題 4，1

問題 1，3 を解くと $x^{13}=(0, 9)$ をえるが，これは明らかに制約条件 4 を侵害している．問題 1，4 よりは $x^{14}=(0.5)$，$\phi=100.0$ をえる．これはどの制約条件も侵害していない．問題 3，1 は実質的に問題 1，3 と同じである．問題 3，4 を解くと $x^{34}=(8, 1)$，$\phi=-568.0$ をえる．これもすべての制約条件を満足している．問題 4，1 は実質的に問題 1，4 と同じである．したがって，ここでえた解のうち実行可能のものは x^{14}, x^{34} の 2 つである．

これについて最後のルールを適用する．これはうえのようにしてえた実行可能の最大化ベクトルを生み出した制約条件集合より 1 制約条件をとり除いて作られた問題の解が，とり除かれた制約条件を侵害することを述べている．

うえの x^{34} については，条件 4 を除いてえる最適解 x^3 は制約条件 4 を侵害しているが，条件 3 を除いてえる最適解 x^4 は制約条件 3 を侵害していない．また x^{14} については，条件 4 を除いてえる最適解 x^1 は制約条件 4 を侵害しており，条件 1 を除いてえる最適解 x^4 は制約　九二
条件 1 を侵害している．それでこのルールは満たされている．それで点 x^{14} が最適解である．[6]

Ⅲ　代数的基礎

タイルとバン・デ・パンの方法はm個の制約条 $C'x \leqq d$ のもとで $\phi(x)$ を最大化するベクトル$\hat{x}$が，この制約条件のうちバインディングな部分集合$\hat{S}$の条件のもとに $\phi(x)$ を最大化することによってえられるという考えにもとづいている．問題はこの部分集合$\hat{S}$——これを解集合 solution set とよぶ——を決定することである．ここでは$\hat{S}$は一意であると仮定する．これについてはタイルとバン・デ・パン自身によっても展開されているが，ここではブートにしたがって述べる．

$\hat{S}$をもとめることがわれわれの課題である．そうであるが，そのまえに，その準備として，ここで等式の形の制約条件の集合Sを条件として $\phi(x)$ を最大化するベクトル x^S をもとめておこう．これは容易にもとめることができる．いまm個の制約条件の配列を変えて，Sに属するものをはじめに述べておく.(7)

目的関数 $\phi(x)=a'x-\frac{1}{2}x'Bx$ を等式の制約条件

$$C'_S x-d=0$$

のもとに最大化するために，ラグランジュ関数

$$a'x-\frac{1}{2}x'Bx-u'_S(C'_S x-d_S)$$

をつくる．ここに u_S はラグランジュ乗数のベクトルである．この式をxに関して微分して，その結果をゼロとおいて

$$x^S=x^\phi-B^{-1}C_S u_S$$

---

(6)　なお，後で用いるラグランジュ乗数の値をもとめておくと，つぎの通りである．

問題3, 4	−299.0	145.
問題1, 4	17.5	7.5

問題3, 4のラグランジュ乗数に負のものがあることに注目しておこう．

(7)　ここでベクトル，行列をつぎのように分割しておく．

$$c'=\begin{bmatrix}c_S{}'\\ c_T{}'\end{bmatrix},\quad d=\begin{bmatrix}d_S\\ d_T\end{bmatrix},\quad u=\begin{bmatrix}u_S\\ u_T\end{bmatrix},$$

をえる．

ここで，この x^S がもとの問題のすべての制約条件を満足しているための条件をもとめておこう．このためには $C'x-d$ すなわち

$$C'_S x^S - d_S$$
$$C'_T C^S - d^T$$

が非負でなければならない．前者は x^S の定義によりゼロ，したがって非負であることが明白である．後者を変形するためにつぎの関係を用いる．すなわちうえの x^S の式の両辺に左より C' を掛けて，行列を分割してあらわすと

$$C'x^S = C'x\phi - C'B^{-1}C_S u_S$$

すなわち

$$\begin{bmatrix} C'_S x^S \\ C'_T x^S \end{bmatrix} = \begin{bmatrix} C'_S x\phi - C'_S B^{-1} C_S u_S \\ C'_T x\phi - C'_T B^{-1} C_S u_S \end{bmatrix} = \begin{bmatrix} d_S + V_S - G_S u_S \\ d_T + V_T - E u_S \end{bmatrix}$$

をえる(8)．これを変形して

$$\begin{bmatrix} V_S - G_S u_S \\ V_T - F u_S \end{bmatrix} = \begin{bmatrix} 0 \\ C'_T x^S - d_T \end{bmatrix}$$

とあらわしておく．このときにはこの第2の要素はつぎのように変形することができる．なお第2の等号においては第1の要素よりえた $u_S = G_S^{-1} V_S$ の関係を用いた．

$$G'x^S - d_T = -F u_S + V_T = -F G_S^{-1} V_S + V_T$$

x^S がもとの問題のすべての制約条件を満足しているためにはこれが非負でなければならない．

(8) ここで行列をつぎのように分割しておく．

$$C'B^{-1}C = \begin{bmatrix} C_S'B^{-1}C_T & C_S'B^{-1}C_T \\ C_T'B^{-1}C_S & C_T'B^{-1}C_T \end{bmatrix} = \begin{bmatrix} G_S & F' \\ F & G_T \end{bmatrix}$$

$$C'x - d = \begin{bmatrix} C_S'x - d_S \\ C_T'x - d_T \end{bmatrix} = \begin{bmatrix} V_S \\ V_T \end{bmatrix}$$

ここで x^s が実行可能性の条件をみたしているか否かの規準をもとめた．この条件をみたしていないものは考慮外におくことができる．今後われわれのとりあげる x^s は実行可能のものとする．(9)

定理　　$u^s>0$ であれば $x^s=\hat{x}$ である．

任意の実行可能ベクトル $\bar{x}$ $(\bar{x} \neq x^s)$ について，つぎの差が正であることを示せばよい．

$$
\begin{aligned}
&\phi(x^s)-\phi(\bar{x})\\
&=\left[a'x^s-\frac{1}{2}(x^s)'Bx^s\right]-\left[a'\bar{x}-\frac{1}{2}\bar{x}'B^{-1}\bar{x}\right]\\
&=a'(x^s-\bar{x})-\frac{1}{2}(x^s)'Bx^s+\frac{1}{2}\bar{x}'B\bar{x}\\
&=a'(x^s-\bar{x})-(x^s)'B\bar{x}+\frac{1}{2}(\bar{x}-x^s)'B(\bar{x}-x^s)
\end{aligned}
$$

B は正値定符号であるから，この式の最後の項は正である．それではじめの 2 つの項が非負であればよい．この項はつぎのように変形できる．

$$
\begin{aligned}
\text{はじめの 2 項}&=(a-Bx^s)'(x^s-\bar{x})\\
&=(u^s)'C'_s(x^s-\bar{x})\\
&=(u^s)'(C'_sx^s-C'_s\bar{x})\\
&=(u^s)'(d_s-C'_s\bar{x})
\end{aligned}
$$

仮定により $u^s>0$ である．また $\bar{x}$ は実行可能であるから $d_s-C'_s\bar{x}\geqq 0$ である．それではじめの 2 項は非負である．

定理　　u^s の要素に負のものがあると $x^s \neq \hat{x}$ である．(10)

証明　　つぎのラグランジュ関数およびその微係数について考えよう．

$$\phi(x, u^s)=a'x-\frac{1}{2}x'Bx-(u^s)'(C'_sx-d_s)$$

---

(9)　退化はないとする．退化の場合はブート〔3〕7.5 節においてとりあげられている．

(10)　命題「もし p であれば q である」の逆の命題は「もし q であれば p である」である．そして後者はその対偶「もし p でなければ q でない」と同じことをあらわす．ここではもとの逆の対偶について考えている．

$$\frac{\partial\phi(x, u^S)}{\partial d_S}=u^S$$

S 内に $\nu_t^S<0$ であるような t があるとする．このときには δ_t を減らすと，$\phi(x^S, u^S)$ を増加させることができることをこの微係数は含意している．ここでつぎのようなバインディングな制約条件の集合について考える．ただしここでの制約条件 t は

$$C_t'x=\delta_t-\varepsilon \qquad \varepsilon>0,\ \varepsilon\to 0$$

の形に変えたものとする．この集合を $S(\varepsilon)$，ここでの最大化ベクトルを $x^{S(\varepsilon)}$ と記す．ここに2つの性質がある．第1に，$x^{S(\varepsilon)}$ よりえられる目的関数の値 $\phi(x^{S(\varepsilon)})$ は，うえで述べたように x^S よりえられるものより大きい．第2に，x^S は実行可能のものである（仮定により）が，このとき x^S を用いると，S 以外の制約条件はすべて若干の余裕をもっている．これらの制約条件はどれも，ϕ を微分小のものとして $x^{S(\varepsilon)}$ によってなお満たされている．それで少なくとも小さい ε について，実行可能ベクトル $x^{S(\varepsilon)}$ で $\phi(x^{S(\varepsilon)})>\phi(x^S)$ が成立するものがある．これは $x^S\neq\hat{x}$ であることを示している．これで定理を証明できた．この証明においては，解ベクトルの連続性の性質により，d が微分小変化しても解集合 $\hat{S}$（バインディングな制約条件のインデックスの集合）は変化しないという事実を用いた．

うえの2つの定理をいっしょに用いると，（退化のないとき）集合 S において，x^S が実行可能のものであるとすると，$u^S>0$ であることが $x^S=\hat{x}$，$S=\hat{S}$ であるための必要かつ十分な条件であることがわかる．

最後に，前節とこの節との関連を述べておこう．前節の末尾でつぎのルールを述べておいた．バインディングな制約条件の集合 S に注目し，これより1つの制約条件を取り除き集合 $S-t$ をつくる．そのときの最大化ベクトルを x^{S-t} として，これが制約条件 t を侵害するか否かをみる．これはスラック $\delta_t-C_t'x^{S-t}$ の正負で示されている．タイルとバン・デ・パンはこのように考えて進む．しかし同じ点はラグ 八八

ランジュ乗数を用いて分析することができる．この後者の方法をブートにしたがってここで述べておいた．スラックを用いる方法とラグランジュ乗数を用いる方法，いいかえるとタイルの方法とブートの方法との関係はつぎの定理に述べられている．

定理　　制約条件の部分集合をSとする．そしてC'_Sがフル横行ランクをもつとする．このSより1つの制約条件tをとり除く．このときにはラグランジュ乗数とスラックのあいだにつぎの関係が成立する．

$$\begin{array}{lll} \nu_t^S>0 & \text{のとき} & \delta_t-C'_t x^{S-t}<0 \\ \quad =0 & \text{のとき} & \qquad\qquad =0 \\ \quad <0 & \text{のとき} & \qquad\qquad >0 \end{array}$$

たとえば最初の関係はつぎのことを述べている．制約条件 $t\in S$ に随伴のラグランジュ乗数が正であるとtを除いた集合 $S-t$ の等式の制約条件のもとでえられる最大化ベクトル x^{S-t} は（いまとり除いた）制約条件tを侵害している．そしてこの逆も成立する．

等式で成立している制約条件の集合をSとする．これに随伴のラグランジュ乗数 u^S は$u^S=G_S^{-1}V_S$ であるから，つぎのようにしてもとめることができる．

$$u^S=(C'_SB^{-1}C_S)^{-1}(C'_Sx-d_S)=P_S^{-1}(C'_Sx-d_S)$$

まずP_Sをつぎのように分割しておく．

$$\begin{aligned} P_S=C'_SB^{-1}C_S&=[C_{S-t}c_t]'B^{-1}[C_{S-t}c_t] \\ &=\begin{bmatrix} C'_{S-t}B^{-1}C_{S-t} & C'_{S-t}B^{-1}c_t \\ c_tB^{-1}C_{S-t} & c'_tB^{-1}c_t \end{bmatrix}=\begin{bmatrix} P_{S-t} & p \\ p' & \pi \end{bmatrix} \end{aligned}$$

そしてこの逆行列をつぎのように表す．

$$P_S^{-1}=\begin{bmatrix} P_{S-t}^{-1}+P_{S-t}^{-1}pp'P_{S-t}\kappa & -P'_{S-t}p\kappa \\ -p'P_{S-t}\kappa & \kappa \end{bmatrix}=\begin{bmatrix} Q & q \\ q' & \kappa \end{bmatrix}$$

八七

$$\kappa=1/(\pi-p'P_{S-t}p)$$

C^1_S がフル横行ランクをもっておれば，この逆行列が存在する．われわれの関心のもっているのは，ここで取り除かれた制約条件tに随伴

のラグランジュ乗数の符号である．これは u_S の式右辺の最後の要素である．そしていまえた逆行列を考慮するとつぎのようにあらわされる．

$$\nu_t^S = -\kappa[p'P_{S-t}^{-1},\ 1]\begin{bmatrix} C'_{S-t}x\phi - d_{S-t} \\ c'_t\ x\phi - \delta_t \end{bmatrix}$$

ここで u^{S-t} の式を用いて書き改めると

$$\nu_t^S = \kappa\{p'u^{S-t} - (c'_t x - \delta_t)\}$$

となる．さらにうえでえた $c'_t x^S = c'_t x\phi - c'_t B^{-1}c_S u_S$ を用いると，この式は $t \notin S-t$ について

$$c'_t x^{S-t} = c'_t x\phi - c'_t B^{-1}C_{S-t}u^{S-t} = c'_t x\phi - p'u^{S-t}$$

が成立することを述べている（ここで p の定義を用いた）．これを用いると

$$\nu_t^S = -\kappa(\delta_t - c'_t x^{S-t})$$

をえる．ここにスカラー κ は正値定符号の行列の逆行列の対角線上の要素であるから正値である．したがって ν_t^S の符号はスラック（$\delta_t - c'_t x^{S-t}$）の符号とちょうど反対である．

最後に集合 $\hat{S}$ をえるために用いる定理をつぎに述べておこう．

定理　$S \subset \hat{S}$ であり，$D = \hat{S} - S$ が空でなければ $C'_D x^S \nleqq d_D$ である．

すなわち，S が $\hat{S}$ の真の部分集合であり，D が $\hat{S}$ 内にあるが，S 内にない制約条件の集合であるとする．そのときには x^S はD内の制約条件の一部を侵害する．

証明　まえに一度用いた関係 $d_S = c'_S x^S = c'_S x\phi - c'_S B^{-1}c_S u_S$ をふたたび利用する．ただしここではSとDに関するものに分けておくとつぎのように記すことができる．

$$(C_S C_D)'B^{-1}(C_S C_D)\begin{bmatrix} u_S^{\hat{S}} \\ u_D^{\hat{S}} \end{bmatrix} = \begin{bmatrix} C'_S x\phi - d_S \\ C'_D x\phi - d_D \end{bmatrix}$$

ここで，$x\phi = B^{-1}C_S u_S + x^S$ を用いると，この右辺はつぎのように書き改められる．

$$右辺=\begin{bmatrix} C'_S B^{-1} C_S u^S + C'_S x^S - d_S \\ C'_D B^{-1} C_D u^S + C'_D x^S - d_D \end{bmatrix}$$

$$=\begin{bmatrix} C'_S B^{-1} C_S u^S \\ C'_D B^{-1} C_D u^S \end{bmatrix} + \begin{bmatrix} C'_S x^S - d_S \\ C'_D x^S - d_D \end{bmatrix}$$

$$=(C_S C_D)' B^{-1} (C_S C_D) u^S - \begin{bmatrix} 0 \\ C'_D x^S - d_D \end{bmatrix}$$

したがって上の式はつぎのように書き改めることができる．

$$(C_S C_D)' B^{-1} (C_S C_D) \begin{bmatrix} u_S^{\hat{S}} - u^S \\ u_D^{\hat{S}} \end{bmatrix} = \begin{bmatrix} 0 \\ C'_D x^S - d_D \end{bmatrix}$$

これに左より〔$(u_S^{\hat{S}} - u^S)\ u_D^{\hat{S}}$〕を掛けると，左辺は正の値となる．行列 $C_{\hat{S}}' B^{-1} C_{\hat{S}}$ が正値定符号だからである．そして右辺は $(u_D^{\hat{S}})'(C'_D x^S - d_D)$ となる．ここに $u_D^{\hat{S}}$ は狭義正値のものである．したがって $C'_D x^S - d_D$ の少なくとも1つの要素が正である．これで定理の証明を終る．それでもし S が $\hat{S}$ の真の部分集合であり，$\hat{S}-S=D$ が空でなければ x^S は D 内の制約条件の少なくとも1つを侵害する．この定理の帰結として――この逆として――x^S によって侵害されている制約条件の少なくとも1つが D 内の制約条件である，をえる．

Ⅳ　数値例，計算方法

ハウタッカーの用いた例をここで利用してタイルーバン・デ・パンによる計算方法を説明しよう．彼はつぎの形の目的関数について考える．

$$\begin{aligned}\phi(x) &= 18\xi_1 + 16\xi_2 + 22\xi_3 + 20\xi_4 \\ &\quad - \frac{1}{2}\{6\xi_1^2 + 2\xi_1\xi_2 + 16\xi_1\xi_3 + 10\xi_2^2 + 2\xi_2\xi_3 + 8\xi_2\xi_4 + 17\xi_3^2 \\ &\quad + 6\xi_3\xi_4 + 11\xi_4^2\}\end{aligned}$$

ここで行列を

八五

$$a=\begin{pmatrix} 18 \\ 16 \\ 22 \\ 20 \end{pmatrix},\quad B=\begin{pmatrix} 6 & 1 & 8 & 0 \\ 1 & 10 & 1 & 4 \\ 8 & 1 & 17 & 3 \\ 0 & 4 & 3 & 11 \end{pmatrix}$$

と定めると，目的関数はまえに述べた形 $\phi(x)=a'x-\frac{1}{2}x'Bx$ となる．これを最大化するにあたって2つのタイプの制約条件がついている．第1は生産数量はどれも負とならないというものである（つぎの式1，2，3，4でこれを示す）．第2は生産要素の利用可能性が限られていることをあらわすものである．たとえば ξ_1 の1単位の生産には生産要素1が5単位必要である．そしてこの要素の利用可能性は2に限られている．また ξ_2 の生産には生産要素2が用いられる．この利用可能は3と限られている．このように2つの付加的制約条件がある．（つぎの式5，6でこれを示す．[11]）

$$\begin{array}{llllll}
1 & -\xi_1 & & & & \leqq 0 \\
2 & & -\xi_2 & & & \leqq 0 \\
3 & & & -\xi_3 & & \leqq 0 \\
4 & & & & -\xi_4 & \leqq 0 \\
5 & 5\xi_1 & & +10\xi_3 & & \leqq 2 \\
6 & & 4\xi_2 & & +5\xi_4 & \leqq 3
\end{array}$$

ここで行列 C', d をつぎのように定める．

$$C'=\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 5 & 0 & 10 & 0 \\ 0 & 4 & 0 & 5 \end{pmatrix},\quad d=\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 3 \end{pmatrix}$$

これを用いると制約条件を $C'x\leqq d$ とあらわすことができる．

なお以下の計算においてはしばしば用いる行列 $C'B^{-1}C$ をここで準備しておく．

$C'B^{-1}C$

(11) ハウタッカー〔5〕は，この他に $\xi_1+\xi_2+\xi_3+\xi_4\leqq 1.666$ の形の制約条件を用いている．ここでは便宜上，この制約条件を省略した．このため計算の手数は約半分になっている．

$$
=\begin{pmatrix}
0.521 & -0.063 & -0.258 & 0.093 & -0.025 & -0.212 \\
-0.063 & 0.124 & 0.012 & -0.054 & -0.003 & -0.228 \\
-0.258 & 0.032 & 0.189 & -0.063 & -0.605 & 0.188 \\
0.093 & -0.054 & -0.063 & 0.127 & 0.166 & -0.422 \\
-0.025 & -0.003 & -0.605 & 0.166 & 6.181 & -0.817 \\
-0.212 & -0.228 & 0.188 & -0.422 & -0.817 & 3.027
\end{pmatrix}
$$

ここで計算手続きを実行可能性の吟味，ラグランジュ乗数の吟味の2つに分けて述べる．

実行可能性の吟味[12]

ラウンド　0

無制約条件のもとに $\phi(x)$ を最大化し x^{ϕ} をもとめる．6つの制約条件のどれも等式の形のものとして用いていないことを

0　0　0　0　0　0

で示す．そしてスラック・ベクトルを計算する．ラウンド0においては，これは

$$-C'x^{\phi}+d=-C'B^{-1}a+d$$

によってもとめられる．もしこのスラック・ベクトルの要素に負のものがあると，つぎのラウンドへ移る．われわれの例においてはつぎのようになる．

4.559　0.474　−1.229　1.980　−8.507　−8.802

ここには x^{ϕ} が制約条件3，5，6を侵害していることがみられる．このインデックスを書き出しておいた．そうしてつぎのラウンドへ移る．

ラウンド　1

ラウンド0において侵害されている制約条件のそれぞれを追加的に制約条件として用いる．例としてその1つ，

3　0　0　0　0　0

(12)　この段階は下記，前段のプログラム横組みのものにおいて詳細に記しておいたのでそれを参照されたい．

の場合について計算の詳細を述べる．もしここでえられるスラック・ベクトルにマイナス符号の要素のないときには残りの段階を省略して，ラグランジュ乗数の吟味のステップへ移る．マイナス要素があるときには5あるいは6を制約条件とする場合について同じように吟味する．そのいずれにおいてもマイナス要素がさけられないときにはつぎのラウンドへ移る．x^3についての計算例をここに述べておく．

$$FG_S^{-1}V_S-V_T=SLACK$$

$$\begin{pmatrix} -0.258 \\ 0.032 \\ 0.000 \\ -0.063 \\ -0.605 \\ 0.188 \end{pmatrix} [0.189]^{-1}[1.229]-\begin{pmatrix} -4.559 \\ -0.474 \\ 0.000 \\ -1.980 \\ 8.507 \\ 8.802 \end{pmatrix}=\begin{pmatrix} 2.886 \\ 0.683 \\ 0.000 \\ 1.569 \\ -12.430 \\ -\ 7.582 \end{pmatrix}$$

ここにえられるスラック・ベクトルの3つ目の要素は0である．制約条件3が等式の形で用いられるためにこのようになっている．そしてx^Sによって侵害されている制約条件のインデックスを末尾に記しておいた．ラウンド0において侵害されている3個の制約条件のそれぞれを用いて，同様の計算をする（前段のプログラムにおいてはこれを省略しておいた）．

ラウンド　2

つぎに適当な2要因集合について考える．この集合は，ラウンド1において等式の形で用いられた制約条件のインデックスに，そのときに侵害されている制約条件のインデックスの1つを付加的に結合してつくる．こうしてえる2要因制約条件のそれぞれについてスラック・ベクトルをもとめる．えられるスラック・ベクトルの要素の中に負のものがなければ，このラウンドのその後の手続きを省略してつぎのステップへ移る．負の要素がなお残っておればラウンド3へ進む．

ここで

3　5　0　0　0　0

についての計算を記すとつぎのようになる.

$$\begin{pmatrix} -0.258 & -0.293 \\ 0.032 & -0.039 \\ 0.088 & 0.000 \\ -0.063 & -0.103 \\ 0.000 & 0.000 \\ 0.188 & 0.675 \end{pmatrix} \begin{bmatrix} 0.189 & 0.099 \\ -0.605 & 6.181 \end{bmatrix}^{-1} \begin{bmatrix} 1.229 \\ 8.507 \end{bmatrix}$$

$$- \begin{pmatrix} -4.559 \\ -0.474 \\ 0.000 \\ -1.980 \\ 0.000 \\ 8.802 \end{pmatrix} = \begin{pmatrix} 0.400 \\ 0.974 \\ 0.000 \\ 1.463 \\ 0.000 \\ -8.217 \end{pmatrix}$$

ここでは制約条件3，5が等式の形で用いられているために，スラック・ベクトルの第3，第5の要素は0となっている．そしてこのベクトルの要素のうち負のものに対応する制約条件のインデックス6を末尾に記しておいた．ここで考慮すべき2要因集合が14個ある．それについてスラック・ベクトルをもとめるとそのどれにも負の要素がある．すなわち2のラウンドのどの最大化ベクトル x^S にも実行可能のものはない．それでラウンド3へ進む．

ラウンド　3

ここで3要因制約条件集合について考える．この集合のそれぞれについてスラック・ベクトルを計算する．ここに負の要素が1つもなければつぎの段階へ進む．もしどのベクトルにも負の要素が含まれておればラウンド4へ進む．そしてそこで4要因制約条件を同じように取扱う．

八一　いまの例においては，考察すべき最初の3要因集合

3　5　6　0　0　0

について計算の詳細はつぎのようになる．

$$\begin{pmatrix} -0.258 & -0.025 & -0.212 \\ 0.032 & -0.003 & -0.228 \\ 0.000 & 0.000 & 0.000 \\ -0.063 & 0.166 & -0.422 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{pmatrix} \begin{bmatrix} 0.189 & -0.605 & 0.188 \\ -0.605 & 6.181 & -0.817 \\ 1.888 & -0.817 & 3.027 \end{bmatrix}^{-1}$$

$$\times \begin{bmatrix} 1.229 \\ 8.507 \\ 8.802 \end{bmatrix} - \begin{pmatrix} -4.559 \\ -0.474 \\ 0.000 \\ -1.980 \\ 0.000 \\ 0.000 \end{pmatrix} = \begin{pmatrix} 0.400 \\ 0.233 \\ 0.000 \\ 0.413 \\ 0.000 \\ 0.000 \end{pmatrix}$$

ここにスラック・ベクトルの要素の中に負のものはない．それでつぎのステップへ移る．

ラグランジュ乗数の吟味

うえのステップにおいて最大化ベクトル x^S がどの制約条件も侵害せず実行可能のものであることがわかるとただちにこの段階へ来る．そしてこのところで等式の形で用いられている制約条件に対応するラグランジュ乗数のベクトルをもとめる．そしてこのベクトルの要素がすべて正であればこの x^S が最適解 $\hat{x}$ である．すなわち $x^S=\hat{x}$ である．もしこのベクトルの要素に負のものがあれば $x^S \neq \hat{x}$ である．このときにはまえの実行可能性の吟味の段階へもどる．

われわれの数値例においては，ラグランジュ乗数のベクトルは，

$$u_S=G_S^{-1}V_S$$

$$\begin{bmatrix} 0.189 & -0.605 & 0.188 \\ -0.605 & 6.181 & -0.817 \\ 4.188 & -0.817 & 3.027 \end{bmatrix}^{-1} \begin{bmatrix} 1.229 \\ 8.507 \\ 8.802 \end{bmatrix} = \begin{bmatrix} 13.407 \\ 3.073 \\ 2.903 \end{bmatrix}$$

と計算される．このベクトルに負のものはない．それでこの x^S が最適解である．(13) この x^S は $x^S=x^\phi-B^{-1}C_Su_S$ を変形した

(13)　この点の吟味の簡便法あるいは x^{S-t} の実行可能性を用いる方法がある．

$$X^S = B^{-1}(A - C_S u_S)$$

によって計算する．これは

$$\begin{pmatrix} 6.000 & 1.000 & 8.000 & 0.000 \\ 1.000 & 10.000 & 1.000 & 4.000 \\ 8.000 & 1.000 & 17.000 & 3.000 \\ 0.000 & 4.000 & 3.000 & 11.000 \end{pmatrix}^{-1} \times \left\{ \begin{pmatrix} 18.000 \\ 16.000 \\ 22.000 \\ 20.000 \end{pmatrix} - \begin{pmatrix} 0.000 & -5.000 & 0.000 \\ 0.000 & 0.000 & 4.000 \\ -1.000 & 10.000 & 0.000 \\ 0.000 & 0.000 & 5.000 \end{pmatrix} \begin{bmatrix} 13.407 \\ 3.073 \\ 2.903 \end{bmatrix} \right\} = \begin{pmatrix} 0.400 \\ 0.233 \\ 0.000 \\ 0.413 \end{pmatrix}$$

となる．このとき目的関数の値は

$$\phi(x^s) = (1/2)(X^s)'(A + C_S u_S)$$

$$= \frac{1}{2} \begin{pmatrix} 0.400 \\ 0.233 \\ 0.000 \\ 0.413 \end{pmatrix}' \left\{ \begin{pmatrix} 18.000 \\ 16.000 \\ 22.000 \\ 20.000 \end{pmatrix} + \begin{pmatrix} 0.000 & -5.000 & 0.000 \\ 0.000 & 0.000 & 4.000 \\ -1.000 & 10.000 & 0.000 \\ 0.000 & 0.000 & 5.000 \end{pmatrix} \begin{bmatrix} 13.407 \\ 3.073 \\ 2.903 \end{bmatrix} \right\}$$

$$= 17.029$$

ともとまる．このプロセスの各ラウンドを述べたものが後段のプログラム（横組みのもの）である．[14]

参 考 文 献

1. Barankin, F. W. and R. Dorfman, "On Quadratic Programming." *University of California Publications in Statistics*, Vol. 2, 1958.
2. Beal, F. M. L., "On Quadratic Programming," *Naval Research Logistic Quarterly*, Vol. 6, 1959, pp. 227–244.
3. Boot, J. C. G., *Quadratic Programming*, 1964.
4. Gue, R. L. and M. E. Thomas, *Mathematical Methods in Operations Research*, 1968.

---

(14)　このプログラムの作成にあたっては安藤助手の助力をえた．なお計算にあたってはコンピューター FACOM 230-10 を用いた．また本学計算センターの TOSBAC-3400 により制約条件式 7 個の場合について計算するのに 45.26秒を要した．

5. Houthakker, H. S., "The Capacity Method of Quadratic Programming," *Econametrica*, Vol. 28, 1960 pp. 62–87.
6. Theil, H. and C. VAN DE Panne, "Quadratic Programming as an Extension of Conveotional Quadratic Maximization," *Management Science*, Vol. 7, 1960, pp. 1–20.
7. Vajda, S., *Mathematıcal Programming*, 1961.

```
COMPILATION BEGIN

PAGE  01

*1004
C       QUADRATIC PROGRAMMING
C       THEIL-VAN DE PANNE PROCEDURE
C       FEASIBILITY TEST
C
        DIMENSION CBC(6,6),V(6),G(6,6)
        DIMENSION GS(6,6),SLACK(6)
        DIMENSION INDEX(6),NEGA(6)
        DO 22 I=1,6
   22   READ(10,101) (CBC(I,J),J=1,6)
        READ(10,101) (V(I),I=1,6)
  101   FORMAT(E17.10)
        NOR=6
C
        NSTEP=0
        WRITE(20,117)
  117   FORMAT(///,14HBINDING CONST.)
        DO 23 I=1,NOR
   23   INDEX(I)=0
        WRITE(20,121) (INDEX(I),I=1,NOR)
  121   FORMAT(6I2)
        DO 24 I=1,NOR
   24   SLACK(I)=V(I)
        GO TO 26
C
C       INDEX OF BINDING CONSTRAINT
   61   NSTEP=NSTEP+1
        INDEX(NSTEP)=NEGA(1)
        WRITE(20,117)
        WRITE(20,121) (INDEX(I),I=1,NOR)
        DO 21 I=1,NO
   21   NEGA(I)=0
C
```

七七

```
C       G,V MATRIX
        WRITE(20,208)
   208  FORMAT(/,10HG,V MATRIX)
        N=NSTEP+1
        DO 33 I=1,NSTEP
        JTHC=INDEX(I)
        DO 33 ITHR=1,NOR
    33  G(ITHR,I)=CBC(ITHR,JTHC)
        DO 34 ITHR=1,NOR
    34  G(ITHR,N)=-V(ITHR)
C       WRITE G,V MATRIX

PAGE 02

        CALL1 200
C
C       GS,VS MATRIX
        WRITE(20,209)
   209  FORMAT(/,12HGS,VS MATRIX)
        DO 6 I=1,NSTEP
        ITHR=INDEX(I)
        DO 6 J=1,N
     6  GS(I,J)=G(ITHR,J)
C
C       WRITE GS,VS MATRIX
        GO TO(11,12,13),NSTEP
    11  DO 15 I=1,NSTEP
    15  WRITE(20,132) (GS(I,J),J=1,N)
        GO TO 9
    12  DO 16 I=1,NSTEP
    16  WRITE(20,133) (GS(I,J),J=1,N)
        GO TO 9
    13  DO 17 I=1,NSTEP
    17  WRITE(20,134) (GS(I,J),J=1,N)
C
```

```
C      F,VT MATRIX   DENOTE F BY G
     9 WRITE(20,210)
   210 FORMAT(/,11HF,VT MATRIX)
       DO 41 JTHC=1,NSTEP
       ITHR=INDEX(JTHC)
       DO 41 J=1,N
    41 G(ITHR,J)=0.0
C      WRITE F,VT MATRIX
       CALL1 200
C
C      LAGRANGE MULTIPLIER  US
       WRITE(20,211)
   211 FORMAT(/,12HUS=(1/GS)*VS)
       DO 35 K=1,NSTEP
       PIVOT=GS(K,K)
       DO 36 J=1,N
    36 GS(K,J)=GS(K,J)/PIVOT
       DO 37 I=1,NSTEP
       IF(I-K) 38,37,38
    38 GSIK=GS(I,K)
       DO 39 J=1,N
    39 GS(I,J)=GS(I,J)-GSIK*GS(K,J)
    37 CONTINUE
    35 CONTINUE
       WRITE(20,131) (GS(I,N),I=1,NSTEP)
   131 FORMAT(F8.3)
C
C      F*US-VT = SLACK
       DO 42 ITHR=1,NOR
       Z=0.0
       DO 43 J=1,NSTEP
    43 Z=Z+G(ITHR,J)*GS(J,N)
    42 SLACK(ITHR)=Z
       DO 45 ITHR=1,NOR
    45 SLACK(ITHR)=SLACK(ITHR)-G(ITHR,N)
C
```

PAGE 03

```
   26 WRITE(20,118)
   118 FORMAT(/,12HSLACK VECTOR)
      WRITE(20,133) (SLACK(I),I=1,3)
      WRITE(20,133) (SLACK(I),I=4,6)
      WRITE(20,119)
  119 FORMAT(/,15HVIOLATED CONST.)
      NO=0
      DO 51 I=1,NOR
      IF(SLACK(I)) 52,51,51
   52 NO=NO+1
      NEGA(NO)=I
   51 CONTINUE
      IF(NO) 56,56,57
   57 WRITE(20,121) (NEGA(I),I=1,NO)
      IF(NSTEP-3) 61,99,99
   56 WRITE(20,123)
  123 FORMAT(5H NONE)
   99 STOP
C
C     WRITE G,V MATRIX
  200 GO TO(1,2,3),NSTEP
    1 DO 5 I=1,NOR
    5 WRITE(20,132) (G(I,J),J=1,N)
      GO TO 4
    2 DO 8 I=1,NOR
    8 WRITE(20,133) (G(I,J),J=1,N)
      GO TO 4
    3 DO 7 I=1,NOR
    7 WRITE(20,134) (G(I,J),J=1,N)
  132 FORMAT(2F8.3)
  133 FORMAT(3F8.3)
  134 FORMAT(4F8.3)
    4 RETURN1
      END

COMPILATION END
```

```
BINDING CONST.
 0 0 0 0 0 0

SLACK VECTOR
     4.559    0.474   -1.229
     1.980   -8.507   -8.802

VIOLATED CONST.
 3 5 6

BINDING CONST.
 3 0 0 0 0 0

G,V MATRIX
  -0.258   -4.559
   0.032   -0.474
   0.189    1.229
  -0.063   -1.980
  -0.605    8.507
   0.188    8.802

GS,VS MATRIX
   0.189    1.229

F,VT MATRIX
  -0.258   -4.559
   0.032   -0.474
   0.000    0.000
  -0.063   -1.980
  -0.605    8.507
   0.188    8.802

US=(1/GS)*VS
   6.481

SLACK VECTOR
   2.886    0.683    0.000
   1.569  -12.430   -7.582
```

```
VIOLATED CONST.
 5 6


BINDING CONST.
 3 5 0 0 0 0

G,V MATRIX
   -0.258   -0.025   -4.559
    0.032   -0.003   -0.474
    0.189   -0.605    1.229
   -0.063    0.166   -1.980
   -0.605    6.181    8.507
    0.188   -0.817    8.802

GS,VS MATRIX
    0.189   -0.605    1.229
   -0.605    6.181    8.507

F,VT MATRIX
   -0.258   -0.025   -4.559
    0.032   -0.003   -0.474
    0.000    0.000    0.000
   -0.063    0.166   -1.980
    0.000    0.000    0.000
    0.188   -0.817    8.802

US=(1/GS)*VS
  15.817
   2.925

SLACK VECTOR
    0.400    0.974    0.000
    1.463    0.000   -8.217

VIOLATED CONST.
 6
```

```
BINDING CONST.
 3 5 6 0 0 0

G,V MATRIX
   -0.258   -0.025   -0.212   -4.559
    0.032   -0.003   -0.228   -0.474
    0.189   -0.605    0.188    1.229
   -0.063    0.166   -0.422   -1.980
   -0.605    6.181   -0.817    8.507
    0.188   -0.817    3.027    8.802

GS,VS MATRIX
    0.189   -0.605    0.188    1.229
   -0.605    6.181   -0.817    8.507
    0.188   -0.817    3.027    8.802

F,VT MATRIX
   -0.258   -0.025   -0.212   -4.559
    0.032   -0.003   -0.228   -0.474
    0.000    0.000    0.000    0.000
   -0.063    0.166   -0.422   -1.980
    0.000    0.000    0.000    0.000
    0.000    0.000    0.000    0.000

US=(1/GS)*VS
  13.407
   3.073
   2.903

SLACK VECTOR
    0.400    0.233    0.000
    0.413    0.000    0.000
```

七一

```
VIOLATED CONST.
 NONE
STOP  0000
```

```
*1004
C      QUADRATIC PROGRAMMING  THEIL-VAN DE PANNE PROCEDURE
C
       DIMENSION A(4),B(4,4),BINV(4,4),C(6,4),CB(6,4),CBC(6,6)
       DIMENSION D(6),V(6),G(6,6),US(6),SLACK(6),INDEX(4)
       DIMENSION NEW(4),NEGA(150),CS(6,4),CSUS(4),XS(4),ACU(4)
       READ(10,101) (A(I),I=1,4)
       READ(10,101) ((B(I,J),I=1,4),J=1,4)
       READ(10,101) ((C(I,J),I=1,6),J=1,4)
       READ(10,101) (D(I),I=1,6)
  101  FORMAT(F8.3)
       WRITE(20,102)
       WRITE(20,103)
  102  FORMAT(/,5X,21HQUADRATIC PROGRAMMING)
  103  FORMAT(5X,28HTHEIL-VAN DE PANNE PROCEDURE)
C
C           PROBLEM
       NOR=6
       WRITE(20,104)
  104  FORMAT(//,17HXXXX PROBLEM XXXX)
       WRITE(20,105)
       WRITE(20,106)
  105  FORMAT(/,5X,36HOBJECTIVE FUNCTION   Q = AX-(1/2)XBX)
```

```
  106 FORMAT(14X,1HA,15X,1HB)
      WRITE(20,107) (A(I),B(I,1),B(I,2),B(I,3),B(I,4),I=1,4)
  107 FORMAT(7X,F8.3,8X,4F8.3)
      WRITE(20,108)
      WRITE(20,109)
  108 FORMAT(/,5X,18HCONSTRAINT    CX..D)
  109 FORMAT(14X,1HC,39X,1HD)
      WRITE(20,111) (C(I,1),C(I,2),C(I,3),C(I,4),D(I),I=1,NOR)
  111 FORMAT(7X,4F8.3,8X,F8.3)
C
      WRITE(20,112)
  112 FORMAT(/,5X,14HC(1/B)C MATRIX)
      DO 1 I=1,4
      DO 1 J=1,4
    1 BINV(I,J)=B(I,J)
      DO 2 I=1,4
    2 BINV(I,I)=BINV(I,I)+1.0
      DO 5 K=1,4
      PIVOT=BINV(K,K)-1.0
      DO 6 J=1,4
    6 BINV(K,J)=BINV(K,J)/PIVOT
PAGE 02
```

```
      DO 7 I=1,4
      IF(I-K) 8,7,8
    8 BIK=BINV(I,K)
      DO 9 J=1,4
    9 BINV(I,J)=BINV(I,J)-BIK*BINV(K,J)
    7 CONTINUE
    5 CONTINUE
      DO 3 I=1,4
    3 BINV(I,I)=BINV(I,I)-1.0
C
      DO 11 I=1,NOR
      DO 11 J=1,4
      Z=0.0
      DO 12 K=1,4
   12 Z=Z+C(I,K)*BINV(K,J)
   11 CB(I,J)=Z
      DO 13 I=1,NOR
      DO 13 J=1,NOR
      Z=0.0
      DO 14 K=1,4
   14 Z=Z+CB(I,K)*C(J,K)
   13 CBC(I,J)=Z
      WRITE(20,113) ((CBC(I,J),J=1,NOR),I=1,NOR)
  113 FORMAT(7X,6F8.3)
```

```
C
C           FEASIBILITY
      WRITE(20,115)
  115 FORMAT(//,21HXXXX FEASIBILITY XXXX)
C
      NSUM=0
      IEND=0
      NSTEP=0
   61 WRITE(20,116) NSTEP
  116 FORMAT(//,9X,5HROUND,I2)
      WRITE(20,117)
      WRITE(20,118)
  117 FORMAT(/,7HBINDING,3X,5HSLACK,40X,8HVIOLATED)
  118 FORMAT(6HCONST.,4X,6HVECTOR,41X,6HCONST.)
C
C     FORM BINDING CONSTRAINT
      IF(NSTEP) 21,21,22
   21 DO 23 I=1,5
   23 INDEX(I)=0
C     COMPUTE SLACK VECTOR (ROUND 0)
      DO 24 I=1,NOR
      Z=0.0
      DO 25 J=1,4
```

```
25 Z=Z+CB(I,J)*A(J)
   V(I)=-Z+D(I)
24 SLACK(I)=V(I)
   GO TO 26
22 IDO=IOPEN
65 INDEX(1)=NEGA(IDO)/1000
   INDEX(2)=(NEGA(IDO)-INDEX(1)*1000)/100
   M=NEGA(IDO)-INDEX(1)*1000-INDEX(2)*100
   INDEX(3)=M/10
   INDEX(4)=M-INDEX(3)*10

PAGE 03
C
C     COMPUTE SLACK VECTOR
      NEXT=NSTEP+1
      DO 33 I=1,NSTEP
      JTHC=INDEX(I)
      DO 33 JTHR=1,NOR
   33 G(JTHR,I)=CBC(JTHR,JTHC)
      DO 34 JTHR=1,NOR
   34 G(JTHR,NEXT)=-V(JTHR)
C     LAGRANGE MULTIPLIER US
      DO 35 JTHC=1,NSTEP
```

```
      JTHR=INDEX(JTHC)
      PIVOT=G(JTHR,JTHC)
      DO 36 J=1,NEXT
   36 G(JTHR,J)=G(JTHR,J)/PIVOT
      DO 37 I=1,NSTEP
      IF(JTHR-INDEX(I)) 38,37,38
   38 K=INDEX(I)
      GKJ=G(K,JTHC)
      DO 39 J=1,NEXT
   39 G(K,J)=G(K,J)-GKJ*G(JTHR,J)
   37 CONTINUE
   35 CONTINUE
      DO 32 J=1,NSTEP
      K=INDEX(J)
   32 US(J)=G(K,NEXT)
C
      DO 41 JTHC=1,NSTEP
      JTHR=INDEX(JTHC)
      DO 41 J=1,NEXT
   41 G(JTHR,J)=0.0
      DO 42 JTHR=1,NOR
      Z=0.0
      DO 43 J=1,NSTEP
```

```
   43 Z=Z+G(JTHR,J)*US(J)
   42 SLACK(JTHR)=Z
      DO 45 JTHR=1,NOR
   45 SLACK(JTHR)=SLACK(JTHR)-G(JTHR,NEXT)
C
C        FIND VIOLATED CONSTRAINT
   26 N=0
      DO 51 L=1,NOR
      IF(SLACK(L)) 52,51,51
   52 N=N+1
      NEW(N)=L
      IF(NSTEP-4) 53,51,51
   53 NSUM=NSUM+1
      IF(NSTEP) 54,54,55
   54 NEGA(NSUM)=L*1000
      GO TO 51
   55 M=L*10**(3-NSTEP)
      NEGA(NSUM)=NEGA(IDO)+M
   51 CONTINUE
C
      IF(N) 56,56,57
   57 WRITE(20,121) (INDEX(I),I=1,4)
      WRITE-(20,122) (SLACK(I),I=1,NOR)
```

```
PAGE 04
      WRITE-(20,121) (NEW(I),I=1,N)
  121 FORMAT(4I2)
  122 FORMAT(6F8.3,1X)
      DO 62 I=1,N
   62 NEW(I)=0
   71 IF(NSTEP) 64,64,63
   63 IDO=IDO+1
      IF(IDO-IEND) 65,65,64
   64 IOPEN=IEND+1
      IEND=NSUM
      NSTEP=NSTEP+1
      IF(NSTEP-5) 61,99,99
   99 STOP
C
   56 WRITE(20,121) (INDEX(I),I=1,4)
      WRITE-(20,122) (SLACK(I),I=1,NOR)
      WRITE-(20,123)
  123 FORMAT(5H NONE)
C
C          LAGRANGEAN
      WRITE(20,125)
```

```
  125 FORMAT(//,20HXXXX LAGRANGEAN XXXX)
C     WRITE  US
      WRITE(20,126) (US(I),I=1,NSTEP)
  126 FORMAT(/,8HLAGRANGE,4F8.3)
      DO 70 I=1,NSTEP
      IF(US(I)) 71,71,70
   70 CONTINUE
C     COMPUTE  XS = (1/B)*(A-CS*US)
      DO 72 I=1,NSTEP
      K=INDEX(I)
      DO 72 J=1,4
   72 CS(I,J)=C(K,J)
      DO 73 J=1,4
      Z=0.0
      DO 74 I=1,NSTEP
   74 Z=Z+CS(I,J)*US(I)
   73 CSUS(J)=Z
      DO 75 J=1,4
   75 ACU(J)=A(J)-CSUS(J)
      DO 76 I=1,4
      Z=0.0
      DO 77 J=1,4
   77 Z=Z+BINV(I,J)*ACU(J)
```

```
   76 XS(I)=Z
      WRITE(20,127) (XS(I),I=1,4)
  127 FORMAT(8HSOLUTION,4F8.3)
C     COMPUTE  Q(XS) = (1/2)*XS*(A+CS*US)
      DO 78 J=1,4
   78 ACU(J)=A(J)+CSUS(J)
      Z=0.0
      DO 79 J=1,4
   79 Z=Z+XS(J)*ACU(J)
      Q=0.5*Z
      WRITE(20,128) Q
  128 FORMAT(8HMAX OF Q,F8.3)
      END


     QUADRATIC PROGRAMMING
     THEIL-VAN DE PANNE PROCEDURE


XXXX PROBLEM XXXX
```

```
OBJECTIVE FUNCTION    Q = AX-(1/2)XBX
         A               B
     18.000            6.000    1.000    8.000    0.000
     16.000            1.000   10.000    1.000    4.000
     22.000            8.000    1.000   17.000    3.000
     20.000            0.000    4.000    3.000   11.000

CONSTRAINT    CX..D
         C                                            D
     -1.000    0.000    0.000    0.000             0.000
      0.000   -1.000    0.000    0.000             0.000
      0.000    0.000   -1.000    0.000             0.000
      0.000    0.000    0.000   -1.000             0.000
      5.000    0.000   10.000    0.000             2.000
      0.000    4.000    0.000    5.000             3.000

C(1/B)C MATRIX
      0.521   -0.063   -0.258    0.093   -0.025   -0.212
     -0.063    0.124    0.032   -0.054   -0.003   -0.228
     -0.258    0.032    0.189   -0.063   -0.605    0.188
      0.093   -0.054   -0.063    0.127    0.166   -0.422
     -0.025   -0.003   -0.605    0.166    6.181   -0.817
     -0.212   -0.228    0.188   -0.422   -0.817    3.027
```

```
XXXX FEASIBILITY XXXX


           ROUND 0

BINDING     SLACK                                                     VIOLATED
CONST.      VECTOR                                                      CONST.
 0 0 0 0     4.559    0.474   -1.229    1.980   -8.507   -8.802    3 5 6


           ROUND 1

BINDING     SLACK                                                     VIOLATED
CONST.      VECTOR                                                      CONST.
 3 0 0 0     2.886    0.683    0.000    1.569  -12.430   -7.582    5 6
 5 0 0 0     4.524    0.469   -2.062    2.209    0.000   -9.928    3 6
 6 0 0 0     3.940   -0.189   -0.681    0.751  -10.885    0.000    2 3 5


           ROUND 2

BINDING     SLACK                                                     VIOLATED
CONST.      VECTOR                                                      CONST.
```

```
 3 5 0 0   0.400   0.974   0.000   1.463   0.000  -8.217  6
 3 6 0 0   3.001  -0.011   0.000   0.609 -13.009   0.000  2 5
 5 3 0 0   0.400   0.974   0.000   1.463   0.000  -8.217  6
 5 6 0 0   3.788  -0.308  -1.694   0.846   0.000   0.000  2 3
 6 2 0 0   3.799   0.000  -0.599   0.600 -10.999   0.000  3 5
 6 3 0 0   3.001  -0.011   0.000   0.609 -13.009   0.000  2 5
 6 5 0 0   3.788  -0.308  -1.694   0.846   0.000   0.000  2 3


        ROUND 3

BINDING  SLACK                                        VIOLATED
CONST.   VECTOR                                         CONST.
 3 5 6 0   0.400   0.233   0.000   0.413   0.000   0.000  NONE


XXXX LAGRANGEAN XXXX

LAGRANGE  13.407   3.073   2.903
SOLUTION   0.400   0.233  -0.000   0.413
MAX OF Q  17.029
STOP  0000
```